



This work is protected by copyright and other intellectual property rights and duplication or sale of all or part is not permitted, except that material may be duplicated by you for research, private study, criticism/review or educational purposes. Electronic or print copies are for your own personal, non-commercial use and shall not be passed to any other individual. No quotation may be published without proper acknowledgement. For any other use, or to quote extensively from the work, permission must be obtained from the copyright holder/s.

SUBMISSION OF THESIS FOR A RESEARCH DEGREE

Part I. DECLARATION by the candidate for a research degree. To be bound in the thesis

Degree for which thesis being submitted

Title of thesis

This thesis contains confidential information and is subject to the protocol set down for the submission and examination of such a thesis.

YES/NO [please delete as appropriate; if YES the box in Part II should be completed]

Date of submission

(Date of submission must comply with Regulation 2D)

Original registration date

Name of candidate

Research Institute

Name of Lead Supervisor

I certify that:

- (a) The thesis being submitted for examination is my own account of my own research
- (b) My research has been conducted ethically. Where relevant a letter from the approving body confirming that ethical approval has been given has been bound in the thesis as an Annex
- (c) The data and results presented are the genuine data and results actually obtained by me during the conduct of the research
- (d) Where I have drawn on the work, ideas and results of others this has been appropriately acknowledged in the thesis
- (e) Where any collaboration has taken place with one or more other researchers, I have included within an 'Acknowledgments' section in the thesis a clear statement of their contributions, in line with the relevant statement in the Code of Practice (see Note overleaf).
- (f) The greater portion of the work described in the thesis has been undertaken subsequent to my registration for the higher degree for which I am submitting for examination
- (g) Where part of the work described in the thesis has previously been incorporated in another thesis submitted by me for a higher degree (if any), this has been identified and acknowledged in the thesis
- (h) The thesis submitted is within the required word limit as specified in the Regulations

Total words in submitted thesis (including text and footnotes, but excluding references and appendices)

Signature of candidate Date

Note

Extract from Code of Practice: If the research degree is set within a broader programme of work involving a group of investigators – particularly if this programme of work predates the candidate's registration – the candidate should provide an explicit statement (in an 'Acknowledgments' section) of the respective roles of the candidate and these other individuals in relevant aspects of the work reported in the thesis. For example, it should make clear, where relevant, the candidate's role in designing the study, developing data collection instruments, collecting primary data, analysing such data, and formulating conclusions from the analysis. Others involved in these aspects of the research should be named, and their contributions relative to that of the candidate should be specified (*this does not apply to the ordinary supervision, only if the supervisor or supervisory team has had greater than usual involvement*).

Dynamical evolution of globular cluster mass functions

Kevin Martin Thomas
B.Sc. Hons

Doctor of Philosophy

Department of Astrophysics, University of Keele.

July 2015

Abstract

Simple single parameter models describing the evolution of globular cluster mass functions (GCMFs) are applied to early type Virgo galaxies. These models assume the dominant form of mass-loss in globular clusters (GCs) is two-body relaxation driven evaporation, and that the cluster initial mass function (CIMF) is described by a Schechter (1976) function. It is concluded that evaporation is primarily responsible for turning a Schechter (1976) CIMF into an evolved GCMF as observed in the Milky Way and other extant galaxies, and an estimate for the corresponding mass-loss rate is made. However, these models do not address the problem of why the GCMF is observed to be the same at all radii, and do not fully recover the shape of the GCMF in the most massive galaxies.

Following this, a method for modelling initial globular cluster systems (GCSs) using quasi-separable distribution functions is described. Quasi-separable distribution functions can be represented as the product of a function of orbital energy, E , and a function of orbital angular momentum, L , i.e. $f(E, L) = g(E)j(h)$ where $h = L/[L_c(E)+L_0]$ with $L_c(E)$ the orbital angular momentum of a circular orbit with orbital energy E and L_0 representing an anisotropy radius times a characteristic velocity. The initial kinematic distribution of the GCS may be specified via the function $j(h)$, known as the circularity function, and the initial spatial distribution, $\nu(r)$, may be specified directly. By definition, the spatial distribution of a system is equal to the integral over all velocities of the distribution function, $\nu = \int_{\text{all } v} g(E)j(h) d^3v$. A technique is described for numerically solving this integral equation for $g(E)$. Once the full distribution function is known, the initial GCS is populated by Monte Carlo sampling of the functions $\nu(r)$, $g(E)$ and $j(h)$ for GC position, orbital energy, and orbital angular momentum for as many globular clusters are desired in the initial GCS. Additionally, initial GC masses are obtained by sampling whichever CIMF is desired.

Examples of the application of this method are given for a wide range of GCS spatial distributions, velocity distributions and mass functions, in a variety of host

galaxies. These presented GCSs serve to demonstrate that the method for setting up an initial GCS works robustly and as intended. Furthermore, the velocity anisotropy profiles corresponding to different circularity functions and L_0 are explored in detail, and a general picture of how velocity anisotropy profiles are related to the circularity functions and L_0 is built.

Subsequently, prescriptions for each of the main GC evolution mechanisms; evaporation, tidal shocking, stellar evolution, dynamical friction, and internal evolution, are either presented or derived. These mechanisms depend on GC properties such as mass and half-mass density, in addition to orbital parameters such as orbital pericentre and radial period. In turn, the orbit (and therefore any related orbital parameters) depends on GC properties through dynamical friction. Thus the evolution of each GC is described by a system of 5 differential equations. A technique is described for taking the initial simulated GCSs, and solving this system of equations for every GC to produce an evolved GCS. The application of the evolution can be conducted in such a way that each mechanism offers the choice of different prescriptions, or can be switched off as desired. Several prescriptions taken from the literature are presented for evaporation, including an additional one from the fits of the single parameter models to Virgo GCMFs.

Demonstrations of the application of this technique for evolving simulated initial GCSs are then presented, with the results compared to what is expected based on the discussion of the relevant evolutionary mechanism. At least one demonstration is given for every mechanism, and the evolution procedure is found to produce results which are generally well understood.

These methods for setting up an initial GCS and subsequently evolving it provide the means for much more sophisticated models than those initially applied to early type Virgo galaxies, and allow the comparison of simulated evolved GCS mass functions, spatial distributions, kinematic distributions, and the fundamental plane with those of extant GCSs as observed in galaxies today. Thus these combined methods constitute models which are far more general than has previously been attempted, and will make possible much more detailed investigations into long standing questions.

Contents

Abstract	iii
1 Evolution of Globular Clusters and the Globular Cluster Mass Function	1
1.1 Introduction	1
1.2 Theorems and Definitions	5
1.2.1 Virial Theorem	5
1.2.2 Relaxation Time	6
1.2.3 Tidal Density	9
1.3 Properties of Globular Clusters and Globular Cluster Systems	15
1.3.1 Fundamental Plane	21
1.3.2 Velocity Distribution	23
1.3.3 Mass Functions	25
1.4 Cluster Destruction Mechanisms	28
1.4.1 Stellar Evolution	28
1.4.2 Dynamical Friction	30
1.4.3 Tidal Shocks	32
1.4.4 Evaporation	33
1.4.5 Core Collapse	40
1.4.6 Destruction Mechanism Overview	43
1.5 Dynamical Evolution of the GCMF	47
2 GCMFs in Early-Type Virgo Galaxies	52
2.1 Introduction	52
2.2 Data	56
2.2.1 Incompleteness	58
2.2.2 Mass-to-Light Ratios	59
2.3 Density-Dependent GCMF Models	61
2.4 Results	65
2.5 Discussion	78
3 Initial Set-up of Simulated GCSs	84
3.1 Introduction	84
3.2 Quasi-Separable Distribution Functions	84
3.3 Model Families	86
3.3.1 Polytropes	88
3.3.2 King Models	91
3.3.3 Dehnen Models	93
3.3.4 Power-law Models	94
3.4 Galaxy Parameters	95

3.4.1	Integral Equations	98
3.5	Tracer System Properties	101
3.5.1	The Circularity Function	101
3.6	Tracer Object Generation	105
3.7	Simulated GCS set-up Code Description	109
4	Examples of Simulated Initial GCSs	112
4.1	Method Verification	112
4.2	Velocity Anisotropy and the Circularity Function	152
5	Dynamical Evolution of Simulated GCSs	160
5.1	Equations of Motion	160
5.2	Prescriptions	162
5.2.1	Evaporation	162
5.2.2	Tidal Shocks	180
5.2.3	Stellar Evolution	183
5.2.4	Dynamical Friction	188
5.2.5	Evolution on the Globular Cluster Fundamental Plane	191
6	Examples of Simulated Evolved GCSs	202
6.1	Simulated GCS Evolution Code Description	202
6.2	Evolved Evaporation Models	208
6.3	Evolved Tidal Shock Models	213
6.4	Evolved Stellar Evolution Models	216
6.5	Evolved Dynamical Friction Models	220
6.6	Evolved GCFP Models	223
7	Summary and Discussion	227
7.1	Summary	227
7.2	Future Work	235
A	Virial Theorem	240
B	Trapezoid Quadrature	242
C	Constant β Circularity Function	244
D	Differential Energy Distribution	245
E	Simulated GCS set-up Code	247
F	Simulated GCS Evolution Code	277
	Bibliography	312

List of Figures

1.1	A schematic of a cluster orbiting in a galaxy, and a particle orbiting the cluster, defining the vector positions \mathbf{r}_s , \mathbf{r}_M and \mathbf{x}_t , and the angular velocity vectors $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}_s$	10
1.2	Plots of the quantity that when negative may turn the tidal density imaginary. This quantity depends on cluster and particle orbits, and therefore several curves are plotted.	14
1.3	Plots of several GC observable and model-fit parameters against GC V-band luminosity, for GCs from many different galaxies.	18
1.4	Logarithmic projected half-light radius (or equivalently effective radius for galaxies) and logarithmic velocity dispersions vs luminosity for GCs and galaxies.	22
1.5	Projected line-of-sight velocity dispersions vs projected galactocentric positions in different colour bands for the M87 GCS.	24
1.6	The cluster mass functions of the Milky Way and Antennae GCSs, demonstrating the difference between dynamically old and young cluster mass functions.	27
1.7	The ratio of half-mass radius, r_h , to tidal radius, r_t , as a function of cluster concentration, $c = \log[r_t/r_0]$, for King (1966) models.	36
1.8	Internal GC half-mass density vs galactocentric position for the Milky Way GCS.	39
1.9	Mass functions of Milky Way GCs of different internal density ranges. The density range in each bin is indicated in the top left of each panel (in units of $M_\odot \text{pc}^{-3}$).	40
1.10	GCMF of an intermediate age GCS divided up into ranges of half-mass density, and projected galactocentric distance.	45
1.11	The survival triangle for Milky Way GCs.	46
2.1	Plots of mass-to-light ratio and F475W - F850LP colour vs metallicity.	60
2.2	Model fits to each division of the GCSs of all 22 Virgo galaxies sorted by density.	66
2.3	Model fits to each division of the GCSs of all 22 Virgo galaxies sorted by galactocentric position.	68
2.4	Model turnovers and modes of GC masses in each division against median GC ρ_h for GC density sorted data, and against median GC R_{gc} for galactocentric position sorted data.	74
2.5	Comparison of GCMF turnovers in each division as calculated with W and LP bandpass data.	77
2.6	GC density against galactocentric position for the GCSs of four out of the 22 AVSVCS galaxies.	79

2.7	GC median density and GCMF turnover-mass vs host galaxy luminosity.	82
3.1	Plot of a series of polytrope model densities with asymptotic power-law solutions, with parameter k drawing nearer to that of a Plummer sphere.	91
3.2	Demonstrations of the behaviour of 5 different circularity functions. . .	104
4.1	Plots of number per unit logarithmic mass and cumulative number against mass of random samplings from a Schechter function. Also displayed are the Schechter function and its cumulative distribution for comparison to the sampled data.	114
4.2	Plots of number per unit logarithmic mass and cumulative number against mass of random samplings from a Gaussian distribution. Also displayed are the Gaussian distribution and its cumulative distribution for comparison to the sampled data.	115
4.3	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $\gamma = 0$ Dehnen model host galaxy with $\tilde{L}_0 = 0$	117
4.4	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $\gamma = 1$ Dehnen model host galaxy with $\tilde{L}_0 = 0$	118
4.5	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $\gamma = 2$ Dehnen model host galaxy with $\tilde{L}_0 = 0$	119
4.6	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $W_0 = 10$ King model host galaxy with $\tilde{L}_0 = 0$	120
4.7	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $W_0 = 15$ King model host galaxy with $\tilde{L}_0 = 0$	121
4.8	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $W_0 = 20$ King model host galaxy with $\tilde{L}_0 = 0$	122
4.9	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $k = 1.5$ polytrope model host galaxy with $\tilde{L}_0 = 0$	123
4.10	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $k = 2$ polytrope model host galaxy with $\tilde{L}_0 = 0$	124
4.11	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $k = 2.5$ polytrope model host galaxy with $\tilde{L}_0 = 0$	125

4.12	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $k = 1.5$ power-law model host galaxy with $\tilde{L}_0 = 0$	126
4.13	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $k = 2$ power-law model host galaxy with $\tilde{L}_0 = 0$	127
4.14	Kinematic, energy and spatial samplings plus number density residuals of a simulated isotropic GCS directly tracing a $k = 2.5$ power-law model host galaxy with $\tilde{L}_0 = 0$	128
4.15	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS directly tracing a $\gamma = 1$ Dehnen model host galaxy with constant tangential anisotropy given by the circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$	131
4.16	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS directly tracing a $W_0 = 20$ King model host galaxy with constant tangential anisotropy given by the circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$	132
4.17	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS directly tracing a $k = 2.5$ polytrope model host galaxy with constant tangential anisotropy given by the circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$	133
4.18	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS directly tracing a $k = 2$ power-law model host galaxy with constant tangential anisotropy given by the circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$	134
4.19	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $\gamma = 1$ Dehnen model host galaxy with constant tangential anisotropy given by the circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$	136
4.20	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $W_0 = 20$ King model host galaxy with constant tangential anisotropy given by the circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$	137
4.21	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $k = 2.5$ polytrope model host galaxy with constant tangential anisotropy given by the circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$	138

4.22	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $k = 2$ power-law model host galaxy with constant tangential anisotropy given by the circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$	139
4.23	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $\gamma = 1$ Dehnen model host galaxy with radial anisotropy given by the circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 0$	142
4.24	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $W_0 = 20$ King model host galaxy with radial anisotropy given by the circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 0$	143
4.25	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $k = 2.5$ polytrope model host galaxy with radial anisotropy given by the circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 0$	144
4.26	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $k = 2$ power-law model host galaxy with radial anisotropy given by the circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 0$	145
4.27	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $\gamma = 1$ Dehnen model host galaxy with radial anisotropy given by the circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 100$	147
4.28	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $W_0 = 20$ King model host galaxy with radial anisotropy given by the circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 100$	148
4.29	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $k = 2.5$ polytrope model host galaxy with radial anisotropy given by the circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 100$	149

4.30	Kinematic, energy and spatial samplings plus number density residuals of a simulated GCS following a $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$ double power-law spatial distribution in a $k = 2$ power-law model host galaxy with radial anisotropy given by the circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 100$	150
4.31	Anisotropy profiles of GCSs directly tracing a $\gamma = 1$ Dehnen model host galaxy, with circularity functions $j(h) = \exp(-ah)$, $j(h) = 1 - h^a[1 - \ln(h^a)]$, and $j(h) = 0.5[1 + h \sin(ah)]$, each with $a = 2, 3, 4$ and $L_0 = 0$	153
4.32	Anisotropy profiles of GCSs directly tracing a $\gamma = 1$ Dehnen model host galaxy, with circularity function $j(h) = \exp[-(a - h)^2/2b^2]$, with $a = 0, 0.5, 1$, and $b = 0.5, 1, 2$, with $L_0 = 0$	155
4.33	Anisotropy profiles of GCSs directly tracing a $\gamma = 1$ Dehnen model host galaxy, with circularity function $j(h) = 0.5[1 + h \sin(ah)]$ and $a = -2, -3, -4$, and $L_0 = 0, 10, 50$	157
5.1	Point mass correction integrals for the Hernquist (1990) model. Figure taken from Gnedin, Hernquist & Ostriker (1999).	182
5.2	Progenitor mass of stars evolving off of the main-sequence as a function of time, $m_{to}(\tilde{t})$. Black points are Miller & Scalo (1979) model data as presented in Chernoff & Weinberg (1990). The red curve is the functional fit to these data, and the green curve is the Lamers, Baumgardt & Gieles (2010) functional fit to the lowest metallicity Hurley, Pols & Tout (2000) model, the data points of which are absent.	186
5.3	Plot of the required ratio of half-mass to tidal densities against number of constituent particles for constant half-mass density in a constantly virialised cluster that is undergoing mass-loss, ignoring the effects of stellar evolution.	199
6.1	Mass function of an initially isotropic GCS with a Schechter CIMF, evolved with a radially independent evaporation prescription.	210
6.2	Mass function of an initially isotropic GCS with a Gaussian CIMF, evolved with a radially dependent evaporation prescription.	212
6.3	Plots of the initial and final mass functions, initial and final velocity anisotropy curves, and half-mass density vs orbital pericentre of an initially isotropic GCS with a Schechter CIMF evolved with tidal shocking.	215
6.4	Mass functions of a Schechter CIMF, and the evolved mass functions it has been evolved into according to two different stellar evolution prescriptions.	217
6.5	The average mass of particles in a GC as a function of time, accounting for the loss in mass as stars expire, and the possible retention of any remnants that may form.	219

6.6	Change in tangential and radial velocity dispersions due to dynamical friction as functions of galactocentric radius. Initially the GCS was isotropic and described by a Schechter CIMF.	221
6.7	Initial and final globular cluster fundamental plane in terms of half-mass radius and mass, evolved with stellar evolution and half-mass density evolution. Initially the GCS was isotropic and described by a Gaussian CIMF.	224

List of Tables

2.1	Details of the models displayed in Figures ?? and ??.	70
2.2	Statistics of 22 galaxy fits for different values of M_{sch} .	71
5.1	Lamers, Baumgardt & Gieles (2010) evaporation model parameters for Roche lobe filling clusters.	173
5.2	Lamers, Baumgardt & Gieles (2010) evaporation model parameters for Roche lobe under-filling clusters.	173
5.3	Breakdown of evaporation mass-loss prescriptions from the literature in terms of coefficient, dependence on mass, cluster structure dependence, assumed host galaxy model, and orbital dependence	176

1 Evolution of Globular Clusters and the Globular Cluster Mass Function

1.1 Introduction

One of the most central topics to astrophysics is stellar evolution. Stars are the host to planetary systems, the building blocks of entire galaxies, and the primary source of chemical enrichment in the Universe. Hence understanding stars has profound consequences for our understanding of the Universe. Humanity has only been studying the heavens in earnest for a few thousand years, while even the most short lived stars have lifetimes of millions of years. Thus fitting together the properties of stars of different metallicities and masses into a consistent pattern is much like trying to extrapolate the entire life of a complete stranger from a single photo. Despite this however, while by no means comprehensive, our understanding of stellar evolution is not without accomplishments. For example, it is known that the rate a star evolves at is strongly dependent upon its mass. Furthermore, it is known that age is quite degenerate with metallicity in stars, meaning that unless it is known a priori what the metallicity of the gas a star formed from was, it is impossible to say how much of the present day elemental abundance of a star is due to evolution and how much was primordial. This degeneracy may be broken if one could view a group of stars with the same initial metallicity and age but with different masses, thus providing an ideal ‘laboratory’ to test theories of stellar evolution. Globular clusters (GCs) are potentially one such ideal environment, though there is now mounting evidence for ‘multiple stellar populations’ in GCs, meaning that there was not just a single star-forming event, but at least one additional burst of star formation at a later time (e.g., Bedin et al. 2004, Piotto et al. 2007).

In any event, the ages of Galactic GCs as inferred from their member stars are invariably at least several billions of years, frequently as much as a Hubble time, and consequently the constituents of GCs must have formed when the Universe was much

younger and thus very different from how it is today. Therefore, GCs must contain information about the early conditions of the universe, and consequences to star formation. Furthermore, most if not all stars are thought to form in associations or groups (Lada & Lada 2003; Larsen 2002a; Larsen 2002b), with the majority of disk and halo stars in the Galaxy today primarily being the aftermath of the destruction of these associations. It is not inconceivable that GCs began as an extension of these associations, still surviving to the present day, and as such understanding the evolution of GCs will itself have consequences for our understanding of star formation and the evolution of much more massive structures. Furthermore, the cores of many Galactic GCs are rich sources of X-rays, possibly originating from mass-transfer between main-sequence stars and compact remnant binary pairs, or alternatively may be one indication for the existence of intermediate-mass black holes, the ‘missing link’ between stellar mass and supermassive black holes. The debate over which of these is the correct interpretation continues to this day (e.g., Bahcall & Ostriker 1975; Bahcall & Wolf 1976; Kong et al. 2010; Pepe & Pellizza 2013). Whichever of these may be correct, GCs certainly contain much interesting physics worthy of study.

Historically, GCs in the Milky Way have been considered separate from open clusters, which typically contain 10^2 to 10^4 M_\odot of stellar material, have core densities and central velocity dispersions in the range of roughly 1 to 100 $M_\odot \text{pc}^{-3}$ and roughly 0.3 to 1 km s^{-1} (Sparke & Gallagher 2007), and limiting radii of roughly 5 to 20 pc. Open clusters are a disk population and are relatively young (< 1 Gyr) in the main, most likely because they are disrupted by gravitational encounters with giant molecular clouds before they can get much older (e.g., Spitzer & Chevalier 1973). They are often difficult to observe, being surrounded by gas, dust and field stars, and so only some 1200 are currently known. However, some estimates put the total number of Galactic open clusters as high as $\sim 10^5$ (e.g., Piskunov et al. 2006).

Globular clusters however are quite different, always very old (typically > 10 Gyr, comparable to a Hubble time), and are associated spatially and kinematically with the Galactic halo or bulge. Typically they contain 10^4 to 10^6 M_\odot of stellar material in a roughly spherical distribution, with core densities and central velocity

dispersions of $\sim 10^3$ to 10^7 $M_\odot \text{pc}^{-3}$ and ~ 0.5 to 10 kms^{-1} , and limiting radii of 20 to 70 pc (e.g., McLaughlin & van der Marel 2005). There are around 150 GCs in the Milky Way (Harris 1996), and similarly old and massive cluster populations are found in the haloes of essentially all galaxies in the local universe, generally in numbers that scale with the total baryonic mass of the galaxy (McLaughlin 1999). For example, M87, the giant elliptical at the centre of the Virgo cluster has a mass of $2.4 \pm 0.6 \times 10^{12} M_\odot$ (Wu & Tremaine 2006) and contains ~ 15000 GCs (e.g., Peng et al. 2008). However, not all galaxies strictly obey this scaling — dwarf ellipticals for example, actually scale inversely (Miller et al. 1998). This is likely because clusters form with the dwarf elliptical in accordance with the scaling, but large amounts of gas for future populations will be driven out of the shallow galactic potential due to supernovae explosions, resulting in fewer stars (and hence lower luminosity), yet the same number of original GCs for observations today (Durrell et al. 1996).

The apparent dichotomy between the typical ages and masses of open and globular clusters was for decades reinforced by the lack of any observed analogues of young globular clusters in nearby galaxies; or more specifically, the apparent absence of any young star clusters with masses $\gtrsim 10^5 M_\odot$. This inevitably led to the conclusion that GC formation was in some way a special process, requiring conditions unique to (or at least more common in) the protogalactic era soon after the Big Bang. Fall & Rees (1988) reviewed theoretical ideas about GC formation with this in mind, putting them into three basic categories:

- Primary formation — Globular clusters formed *before* galaxies. For example, Peebles & Dicke (1968) noted that the Jeans mass (the mass at which a cloud of gas at a certain temperature will collapse under its own gravity) at recombination was $\sim 10^6 M_\odot$ and thus suggested that GCs are essentially the result of standard structure formation.
- Secondary formation — Globular clusters formed *with* galaxies. For example, Fall & Rees (1985) showed that thermal instability in the hot, protogalactic

medium would lead to the development of cold pockets of gas with masses $\sim 10^6 M_\odot$, which could then collapse and fragment to form GCs.

- Tertiary formation — Globular clusters formed *after* galaxies. For example, Schweizer (1986) suggested that the strong shocking of molecular gas clouds during mergers of disk galaxies should lead to the copious formation of star clusters with masses ranging up into the GC regime.

Of these possibilities, secondary formation was generally favoured, because there is no evidence for dark matter in GCs (as would be expected for primary formation) and because there were no observations of young GC mass clusters (as expected in tertiary formation). This swiftly changed with the launch of the Hubble Space Telescope (HST), where observations of NGC 1275 by Holtzman et al. (1992) revealed the presence of young, blue clusters with masses $M \sim 10^5$ to $10^8 M_\odot$ forming out of a cooling flow onto this galaxy. However, it was the seminal imaging of the Antennae galaxies (NGC 4038/4039) by Whitmore & Schweizer (1995) that finally saw GC sized gravitationally bound clusters forming in an ongoing merger. Subsequently, “young globulars” have been found in many other local mergers and starbursts (e.g., Larsen & Richtler 1999; Meurer 2000; de Grijs et al. 2003; Anders et al. 2004; Barmby et al. 2009; Bastian et al. 2013).

The reason that such young massive clusters (YMCs) form in mergers and starbursts rather than in quiescent galaxies such as the Milky Way, presumably has to do with the vastly greater abundance of gas in the former systems (e.g., Larson 1993). Considering the Cluster Initial Mass Function (CIMF) to be a probability distribution, a greater gas abundance allows a greater number of random samples from this probability distribution, and hence will lead to a greater number of massive stellar systems. Additionally, Larsen & Richtler (2000) observed that the star formation rate depends on available gas content, so that for a sufficiently high star formation rate, populations of stellar objects may form significantly more quickly than they are destroyed by the death of massive stars or tidal shocks by progenitor gas clouds. Thus given the greater abundance of young clusters and the fact that being young, they contain bright mas-

sive stars, very massive young clusters are more likely to be observed in starbursts. Furthermore, upheaval in the galactic potential during a merger or a close encounter can allow structures much more massive than usual to form (e.g., Renaud et al. 2009).

At this point it is obvious to ask, given a Hubble time, might YMCs evolve into GCs? One way to address this question is to look at individual GC properties, and Globular Cluster System (GCS) properties. A discussion of observable GC properties such as sizes and masses will be given at the beginning of section 1.3, before moving on to a description of observable GCS properties, namely the fundamental plane, GCS kinematics, and the Globular Cluster Mass Function (GCMF). Of these three GCS observables, the GCMF is by far the most widely accessible in other galaxies. Therefore, the main thrust of this thesis will be focusing on the GCMF (introduced in section 1.3.3), and the cluster destruction mechanisms that may turn an assumed CIMF into a GCMF as observed today through mass-loss in constituent GCs (section 1.4). Central to the theory of mass-loss and evolution of GCs are the virial theorem, relaxation, and tidal limitation. Therefore, these are described and derived next.

1.2 Theorems and Definitions

In order to give a coherent and clear description of the properties of, and ongoing processes in, GCs it is necessary to first derive some theorems and quantities, as follows.

1.2.1 Virial Theorem

Many results and other theorems throughout this thesis rely upon the virial theorem, which relates the time averaged total kinetic energy of member stars to the time averaged total binding energy for a stable system (see appendix A):

$$\left\langle \frac{d^2 I_{tot}}{dt^2} \right\rangle = 4 \langle K_{tot} \rangle - 2n \langle W_{tot} \rangle = 0 \quad (1.1)$$

where $\langle x \rangle$ denotes the time average of x , I_{tot} is the total system moment of inertia, K_{tot} is the total internal kinetic energy, W_{tot} is the total binding energy, and for binding energy provided by gravitating point masses, $n = -1$. When a system satisfies equation (1.1) it is said to be in virial equilibrium. Otherwise, it must be an unstable system, $\left\langle \frac{d^2 I_{tot}}{dt^2} \right\rangle \neq 0$. Additionally, the gravitational radius is defined as $r_g \equiv -GM^2/W_{tot}$, where M is the total mass. However, the much more easily obtained radius containing half of the total system mass, r_h , is closely related to r_g for many common models used for stellar systems, with $r_h/r_g \simeq [0.4, 0.5]$ (e.g., King 1966; Hernquist 1990; Jaffe 1983). Thus the total binding energy is often approximated as $W_{tot} \simeq -0.45GM^2/r_h$.

1.2.2 Relaxation Time

Given a bound system of gravitationally interacting particles isolated from any external influence, these particles will share and redistribute their energies with each encounter, such that the distribution of speeds is constantly tending towards a distribution such as a Maxwell-Boltzmann distribution, $N(v) \propto v^2 \exp(-v^2/2\sigma_0^2)$ where σ_0 is some characteristic speed of the system. This means that given enough time, a few particles will have their speeds scattered up to very high values, allowing them to escape the gravitational potential of the system, thus reducing its mass and total energy. Once escaped, the energy distribution will lack the ‘high energy tail’, and will thus begin to relax towards a Maxwell-Boltzmann distribution once again. Hence relaxation is an ongoing process, and is the time for a system to reorganise itself, or alternatively, the time for an average particle to have its velocity altered by order itself.

A particle can have its velocity altered by an amount of order itself by three possible mechanisms: direct collisions, strong encounters, and weak encounters. The time-scale for direct collisions in a system is the mean free time of stars, which is itself just the ratio of the mean free path to the average speed of particles. The reciprocal of the mean free path is estimated as the number density of particles times the cross section of interaction, $1/d = n\pi(2R)^2$, where n is the number density, and $\pi(2R)^2$ is the collisional cross section for two stars each of radius R . The average relative speed

of particles in a system is the velocity dispersion, σ , and thus:

$$t_{coll} = \frac{d}{\sigma} = \frac{1}{4n\pi R^2 \sigma} \simeq 15 \times 10^6 \text{Gyr} \left(\frac{\sigma}{10 \text{ km s}^{-1}} \right)^{-1} \left(\frac{n}{10^3 \text{ pc}^{-3}} \right)^{-1} \left(\frac{R}{R_\odot} \right)^{-2} \quad (1.2)$$

Thus the relaxation time due to direct collisions for a typical GC with $n \simeq 10^3 \text{ pc}^{-3}$, $\sigma \simeq 10 \text{ km s}^{-1}$ and $R \simeq 1 R_\odot$ is of order a million Hubble times.

A strong encounter is where two stars each of mass m with relative speed σ approach within a distance r of one another small enough that the potential energy between them is greater than the kinetic energy of their relative motion:

$$\frac{Gm^2}{r} \gtrsim \frac{1}{2}m\sigma^2 \Rightarrow r \lesssim \frac{2Gm}{\sigma^2} \equiv r_s$$

Then for relative speed σ through a region of space with number density n of stars, for a single strong encounter during a time t_{str} , there must be one other star in the cylinder of space swept out by the subject star's motion. This cylinder has a cross section of πr_s^2 and a length of σt_{str} , and thus $n\pi r_s^2 \sigma = 1$ to have a single strong encounter. Therefore,

$$t_{str} = \frac{1}{n\pi r_s^2 \sigma} = \frac{\sigma^3}{4\pi G^2 n m^2} \simeq 166 \text{Gyr} \left(\frac{\sigma}{10 \text{ km s}^{-1}} \right)^3 \left(\frac{n}{10^3 \text{ pc}^{-3}} \right)^{-1} \left(\frac{m}{0.5 M_\odot} \right)^{-2} \quad (1.3)$$

Thus the relaxation time due to strong encounters for a typical GC with $m \simeq 0.5 M_\odot$, $\sigma \simeq 10 \text{ km s}^{-1}$ and $n \simeq 10^3 \text{ pc}^{-3}$ is of order 10 Hubble times.

Alternatively, a particle may have its square-speed altered by order itself via many distant nudges from other stars. Thus, the relaxation time due to weak encounters depends on the average nudge to a star's kinetic energy by an encounter with another, i.e. $\delta\sigma \propto m/b^2 \times b/\sigma$. This is essentially the acceleration at closest approach of the perturbed star to the perturber times the duration of the encounter, with m the mass of a cluster star, σ the speed of a cluster star, and b the impact parameter (the perpendicular distance between the unperturbed particle trajectory and the perturber). Each star will undergo many such encounters, and so an estimate for the number of these will be given by the surface density of the cluster times the cross sectional area

of a star's interaction sphere: $\delta n = N/\pi r^2 \times 2\pi b \delta b$, where N is the number of stars in the cluster and r is its radius. After one cluster crossing time (within a factor of order unity this is the same as the free-fall time for a cluster star, $t_c \propto (G\rho)^{-1/2}$), the sum of these nudges amounts to:

$$\sum \delta\sigma^2 \simeq \delta\sigma^2 \delta n \propto (m^2 N / b \sigma^2 r^2) db$$

This integrated over the range of impact parameters, b_{min} to b_{max} , then gives the mean deflection in a star's square speed over a single cluster crossing, and also introduces the Coulomb logarithm, $\ln \Lambda = \ln(b_{max}/b_{min})$. The minimum impact parameter is simply the value at the transition from weak encounter to strong encounter, $b_{min} = r_s = 2Gm/\sigma^2$. However, there is much uncertainty in the correct choice for b_{max} , and is generally taken to be of order the size of the system. Assuming that the cluster is virialised, then $\sigma^2 \simeq GNm/r$, and consequently $b_{max}/b_{min} \simeq r\sigma^2/2Gm \simeq 0.5N = \lambda N$, with λ soaking up all the uncertain factors used to arrive at this result. Studies have been conducted treating λ as a fitting parameter to ascertain exactly what functional dependence it should have or what value it should take. These studies concluded that the value of λ depends on whether a cluster is populated only by stars with the same mass or not, with $\lambda \simeq 0.4$ for single-mass clusters, and $\lambda \simeq 0.02$ for multi-mass clusters (Giersz & Heggie 1994; Giersz & Heggie 1996). The mean deflection of a typical star square-speed per crossing time then, is given by:

$$\Delta\sigma^2 \propto \frac{m^2 N}{\sigma^2 r^2} \ln(\lambda N)$$

Since $\sigma^2/\Delta\sigma^2$ is the number of nudges per cluster crossing required to change a typical cluster star square-speed by order of itself, the relaxation time is given by $t_r \propto t_c \times \sigma^4 r^2 / Nm^2 \ln(\lambda N)$ where $t_c \propto r/\sigma$ is the cluster crossing time. Then (e.g., Spitzer 1987):

$$t_r \propto \frac{\sigma^3 r^3}{m^2 N \ln(\lambda N)} \propto \frac{\sigma^3}{m \rho \ln(\lambda N)}$$

Apart from the presence of the Coulomb logarithm, the functional dependence of this equation is the same as that of equation (1.3), and thus for a typical GC with $N \simeq 10^5$, is about an order of magnitude smaller — i.e. many weak interactions are much more important to the dynamical evolution of a populous system of stars than few strong interactions.

At the radius containing half of the cluster mass, r_h , the density is $\rho_h = 3mN/8\pi r_h^3 = 3M/8\pi r_h^3$ and a typical star will have $\sigma^3 \propto (M/r_h)^{3/2} \propto (Nm/r_h)^{3/2} \propto Nm\rho_h^{1/2}$ by the virial theorem. Since the relaxation time depends on local density, it will clearly not be constant across an entire cluster, and consequently is often evaluated at the half-mass radius r_h . Then finally, the scalings for the half-mass relaxation time are obtained. With a more careful derivation, the numerical coefficient may also be brought forward (e.g., Spitzer 1987; Heggie & Hut 2003; Binney & Tremaine 2008):

$$t_{rh} \simeq \frac{2.69 \text{Gyr}}{\ln(\lambda N)} \left(\frac{M}{10^5 \text{M}_\odot} \right) \left(\frac{10^3 \text{M}_\odot \text{pc}^{-3}}{\rho_h} \right)^{1/2} \quad (1.4)$$

Thus the relaxation time due to weak encounters for a typical GC with $M \simeq 10^5 \text{M}_\odot$, $m \simeq 0.5 \text{M}_\odot$ and $\rho_h \simeq 10^3 \text{M}_\odot \text{pc}^{-3}$, is of order 300 Myr. As the time-scale for dynamical evolution of any cluster is just its relaxation time, equation (1.4) is very useful for further investigating cluster evolution. However dynamical evolution of mass functions is brought about primarily by a combination of relaxation and tides. Therefore the next tool required is an expression for the tidal density of clusters.

1.2.3 Tidal Density

The evolution of GCSs is determined by mass-loss in individual GCs, and the rate of mass-loss in individual GCs is determined by their tidal density (see section 1.4). This is the mass content within a specific volume (known as the Roche sphere or lobe) required for a cluster to gravitationally dominate that volume. Outside of the Roche sphere, particles would orbit the host galaxy rather than the cluster. The edge of this Roche sphere is known as the tidal radius, and is the point at which gravitational,

centripetal, and Coriolis forces between the cluster and host galaxy balance. For a cluster specifically on a circular orbit, the tidal radius is also called the Jacobi radius. Thus, once particles associated with a cluster pass beyond the tidal radius, they are lost from the cluster and instead become associated with the host galaxy. Figure 1.1 displays a schematic diagram of the problem set-up.

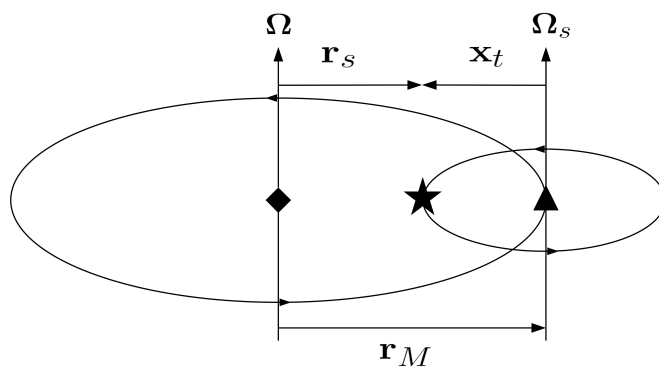


Figure 1.1: A schematic of the system under consideration. Both the cluster (triangle) and particle (star) orbit in the same plane with angular velocity Ω and Ω_s . The problem is considered from a frame centred on the host galaxy (diamond) rotating with angular velocity $\Omega(t)$, such that the cluster centre-of-mass is stationary.

Following Read et al. (2006b), the tidal density can be calculated as follows. In a frame of reference rotating with angular velocity Ω centred on the host galaxy such that the centre-of-mass of the subject cluster is stationary, the equation of motion for the centre-of-mass of the cluster is given by:

$$\ddot{\mathbf{r}}_M + \dot{\Omega} \times \mathbf{r}_M + 2\Omega \times \dot{\mathbf{r}}_M + \Omega \times (\Omega \times \mathbf{r}_M) + \nabla \Phi_g(\mathbf{r}_M) = \mathbf{0} \quad (1.5)$$

where \mathbf{r}_M is the galactocentric position of the cluster, and Φ_g is the host galaxy potential. The first term is the acceleration, the second and fourth terms are centrifugal

effects due to a rotating frame and non-circular orbits, the third term is the Coriolis term due to the rotating frame, and the fifth term is due to the gravity of the host galaxy. The equation of motion of a particle orbiting the cluster in this frame is given by:

$$\ddot{\mathbf{r}}_s + \dot{\boldsymbol{\Omega}} \times \mathbf{r}_s + 2\boldsymbol{\Omega} \times \dot{\mathbf{r}}_s + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r}_s) + \nabla\Phi_g(\mathbf{r}_s) + \nabla\Phi_M(\mathbf{x}_t) = \mathbf{0} \quad (1.6)$$

where \mathbf{r}_s is the galactocentric position of the particle, Φ_M is the cluster potential, and $\mathbf{x}_t = \mathbf{r}_s - \mathbf{r}_M$ is the vector position of the particle relative to the cluster. Considering only the case where the particle is at the instantaneous tidal radius, co-planar (both orbiting on the same plane) radial or circular particle orbits satisfy

$$\dot{\mathbf{r}}_s = \dot{\mathbf{r}}_M + \boldsymbol{\Omega}_s \times \mathbf{x}_t \quad (1.7)$$

where $\boldsymbol{\Omega}_s$ is the angular velocity of the particle around the cluster. In the case of circular orbits, this holds because the velocity of the particle is due only to its own rotation about the cluster, $\boldsymbol{\Omega}_s$, and the motion of the cluster about the host galaxy, $\dot{\mathbf{r}}_M$. For radial orbits, this holds because $\boldsymbol{\Omega}_s = \mathbf{0}$, and as the particle is at apocentre, the radial component of its velocity must also be zero.

Substituting (1.7) into (1.6) and subtracting (1.5) from the resulting equation, the following is obtained:

$$\begin{aligned} \ddot{\mathbf{x}}_t &= \mathbf{F}_f + \mathbf{F} \\ \mathbf{F}_f &= \dot{\boldsymbol{\Omega}} \times \mathbf{x}_t + 2\boldsymbol{\Omega} \times (\boldsymbol{\Omega}_s \times \mathbf{x}_t) + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{x}_t) \\ \mathbf{F} &= \nabla\Phi_g(\mathbf{r}_s) + \nabla\Phi_M(\mathbf{x}_t) - \nabla\Phi_g(\mathbf{r}_M) \end{aligned}$$

and $\ddot{\mathbf{x}}_t = \mathbf{0}$ at the tidal radius by definition. Then, considering only the components of force acting parallel to \mathbf{r}_M ($\mathbf{F}_f \cdot \hat{\mathbf{r}}_M + \mathbf{F} \cdot \hat{\mathbf{r}}_M = 0$):

$$\mathbf{F} \cdot \hat{\mathbf{r}}_M = \left. \frac{d\Phi_g}{dr} \right|_{r=r_M-r_t} - \left. \frac{d\Phi_g}{dr} \right|_{r=r_M} - \left. \frac{d\Phi_M}{dx} \right|_{x=r_t}$$

$$\mathbf{F}_f \cdot \hat{\mathbf{r}}_M = 2\alpha\Omega_s\Omega r_t + \Omega^2 r_t$$

where $\alpha\Omega_s$ is the z component (perpendicular to the orbital plane) of $\mathbf{\Omega}_s$, with $\alpha = 1$ for a prograde circular particle orbit, $\alpha = 0$ for a radial particle orbit, and $\alpha = -1$ for a retrograde circular particle orbit. Ω is the z component of $\mathbf{\Omega}$, and $-r_t$ is the component of \mathbf{x}_t parallel to \mathbf{r}_M (the negative sign is due to the vector orientation, see Figure 1.1). Since different α corresponds to tidal radii for particles on different orbits, the most and second most tightly bound particles will have their tidal radii inside the tidal radius for the least tightly bound particles, and therefore there will be matter outside their tidal radii. Thus it is necessary to assume that clusters are sufficiently centrally concentrated as to be well approximated by a point mass potential at the distance of r_t for these most tightly bound particles. Also employing the distant tide approximation (assuming $r_t \ll r_M$, where r_M is the magnitude of \mathbf{r}_M), it is then possible to show that:

$$\begin{aligned} \mathbf{F} \cdot \hat{\mathbf{r}}_M &= -r_t \left. \frac{d^2\Phi_g}{dr^2} \right|_{r=r_M} - \frac{GM_c}{r_t^2} \\ \mathbf{F}_f \cdot \hat{\mathbf{r}}_M &= \Omega^2 r_t + 2\alpha\Omega \sqrt{\frac{GM_c}{r_t}} \end{aligned}$$

$$0 = \mathbf{F}_f \cdot \hat{\mathbf{r}}_M + \mathbf{F} \cdot \hat{\mathbf{r}}_M$$

which is a quadratic in $\sqrt{GM_c/r_t^3}$, and thus the solution can be written as:

$$\rho_t = \frac{3}{4\pi G} \left[\alpha\Omega \pm \left(\Omega^2(\alpha^2 + 1) - \left. \frac{d^2\Phi_g}{dr^2} \right|_{r=r_M} \right)^{1/2} \right]^2 \quad (1.8)$$

i.e. the tidal density (and therefore rate of mass-loss, see section 1.4) depends on the orbit of the particle in question. Co-planar particle orbits mark the extrema of this effect — prograde particle orbits ($\alpha = 1$) are most easily stripped, while retrograde particle orbits ($\alpha = -1$) are least easily stripped (e.g., Giersz & Heggie 1997; Fukushige & Heggie 2000; Baumgardt & Makino 2003; Read et al. 2006b). Consequently, systems undergoing tidal stripping show tangential velocity anisotropy near their tidal radii (e.g., Kazantzidis, Magorrian & Moore 2004; Kravtsov, Gnedin & Klypin 2004). Setting $\alpha = 0$ returns equation (1.8) to the familiar form e.g. as derived by von Hoerner (1957) and King (1962). Thus for a specific particle orbit and at a specific galactocentric orbit, the tidal density is constant. The tidal radius is derived simply by combining the present cluster mass with the tidal density. As particles pass beyond the tidal radius, they are stripped from the cluster, and consequently the cluster mass decreases causing the tidal radius to shrink.

Read et al. (2006b) studied the evolution of tidal radii with $\alpha = [-1, 0, 1]$ in a dissolving cluster, and found that for circular cluster orbits all particle orbits begin to be stripped according to the prograde particle tidal limit ($\alpha = 1$) at times $t \gtrsim 2$ Gyr. The reason for this is that, even for a spherical cluster potential, the presence of a host galaxy potential permeating the cluster causes the total potential experienced by particles to be non-spherical, and thus the orbital plane of particles precess. Precession can then put a particle onto a more easily stripped orbit, at which point it is carried away by tides. This effect would presumably still occur for clusters on non-circular orbits, but the precession is drowned out by particle orbit migration brought about by the non-uniform tidal field (Read et al. 2006b).

Furthermore, for the tidal density to be real, it is required that the argument of the square root in equation (1.8) satisfies $\Omega^2(\alpha^2 + 1) - \frac{d^2\Phi_g}{dr^2}\bigg|_{r=r_M} \geq 0$. In general, at any radius the angular velocity will be bounded between the angular velocity of a radial orbit and that of a circular orbit; $0 \leq \Omega^2 \leq (4\pi G/3)\bar{\rho}(r_M)$ where $\bar{\rho}(r_M)$ is the average galaxy density inside of r_M , and thus:

$$\frac{2\bar{\rho}(r_M)}{3\rho(r_M)} - 1 \leq \frac{1}{4\pi G\rho(r_M)} \left[\Omega^2(\alpha^2 + 1) - \frac{d^2\Phi_g}{dr^2} \Big|_{r=r_M} \right] \leq \frac{(\alpha^2 + 3)\bar{\rho}(r_M)}{3\rho(r_M)} - 1 \quad (1.9)$$

The isothermal sphere has a distribution function and asymptotic density limits

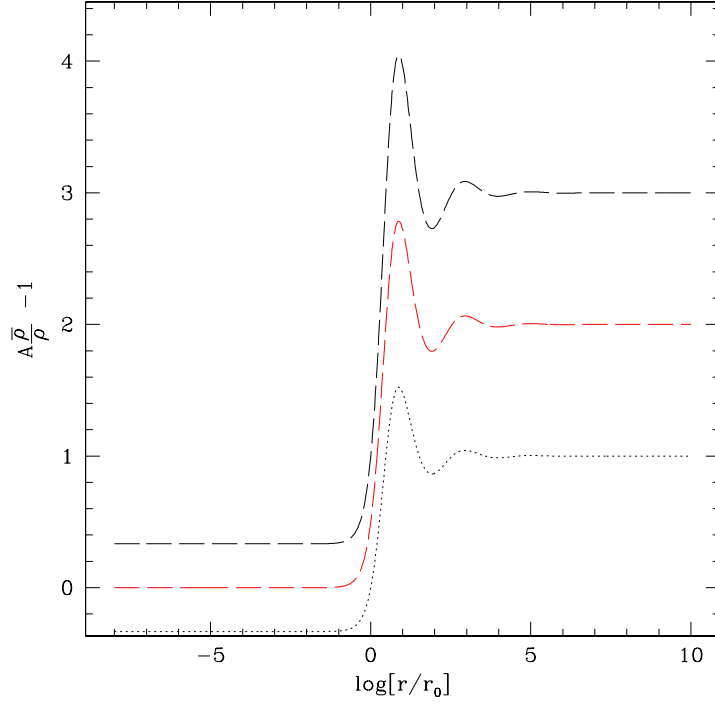


Figure 1.2: Extremes of argument of the square root of equation (1.8) in an isothermal sphere. Long dashed lines are circular cluster orbits, black long dashed is for a circular particle orbit ($\alpha^2 = 1$, $A=4/3$), red long dashed is for a radial particle orbit ($\alpha^2 = 0$, $A=1$). The dotted line is for a radial cluster orbit, in which case the particle orbit no longer matters ($A=2/3$). Also note r_0 is the galactic core radius.

given by:

$$f(E) \propto \exp(-E/\sigma_0^2) \quad (1.10)$$

$$\rho(r) \propto \begin{cases} \text{const} & \text{for } r \ll r_0 \\ r^{-2} & \text{for } r \gg r_0 \end{cases} \quad (1.11)$$

where r_0 is the core radius. Taking this as a model for the host galaxy, Figure 1.2 displays the upper and lower limits of equation (1.9) in the isothermal sphere. As there is also dependence on α to consider, there are two curves for the upper limit, one with $\alpha = 0$, giving a coefficient on $\bar{\rho}(r_M)/\rho(r_M)$ of $A = 1$, and the other curve with $\alpha = 1$ or $\alpha = -1$, giving $A = 4/3$. The lower limit of equation (1.9) has no dependence on α , and thus there is only one curve, with $A = 2/3$. Thus all clusters on any orbit in such a system must fall somewhere between one of the two uppermost curves and the lowermost curve, depending on the particle orbit under consideration. The quantity in equation (1.9) for circular cluster orbits with radial particle orbits (red curve, $A = 1$) tends to zero at small radii, and so since this is a minimum, all clusters on circular orbits will have well defined tidal densities at all radii (excepting of course when $r_M \sim r_t$ i.e. when the distant tide approximation breaks down), irrespective of particle orbits. However, all clusters on non-circular orbits have the quantity in equation (1.9) turn negative roughly inside of the galactic core ($r \lesssim r_0$), also irrespective of particle orbits. The tidal density turning imaginary in this region is in agreement with Dekel, Devor & Hetzroni (2003), who found that tides grow progressively weaker with shallower local density gradients, and in fact turn compressive along the line of centres joining the host galaxy and cluster in constant density environments.

While the derivation of equation (1.8) allows for use on non-circular cluster orbits, care must be taken when doing so. This is because while the structure of a cluster will always attempt to adapt to its imposed tidal density, if the tidal density is changing due to being on a non-circular orbit, the cluster will attempt to adapt to a different structure from one moment to the next. Thus the evolution of a cluster will depend on whether the time-scale for a cluster to reach some sort of equilibrium with its surroundings is greater or less than the time-scale of an orbit. The time-scale for reaching this equilibrium is the relaxation time, typically several Gyr, while a radial period is typically a few hundred Myr. For this reason, von Hoerner (1957) argued that the effective tides (the tidal density matching the actual rate of mass-loss of a cluster, see section 1.4) will be at orbital pericentre — i.e. once a cluster passes pericentre where the tides are strongest, its structure will be trimmed to that limit, and will not have

time to adjust to anything else before the next pericentric passage. While this picture is basically correct, it overlooks an additional complication, namely that as a cluster recedes after a pericentric passage, its tidal radius will grow and may re-encompass some of the particles previously shed. These particles may then become rebound, and thus the rate of cluster mass-loss will be less severe than would be predicted by pericentric effective tides, meaning that the galactocentric distance corresponding to effective tides, r_{ev} , lies further out (e.g., Odenkirchen et al. 1997; Brosche, Odenkirchen & Geffert 1999; Küpper et al. 2010; Küpper, Lane & Heggie 2012; Webb et al. 2013).

1.3 Properties of Globular Clusters and Globular Cluster Systems

The ensemble of an entire population of GCs in a host galaxy is called a globular cluster system. There are a few main observable properties of GCSs that will be focused on here. There is the frequency of GC luminosities in a system, the GC luminosity function (GCLF). Provided a working understanding of what causes mass-loss in clusters, it is possible to take the typical observed luminosity function of YMCs and evolve it for 13 billion years for comparison to GCLFs as observed today, to test the hypothesis that the luminosity function of YMCs today are the analogues of GCs 13 Gyr ago.

Additionally, there is the kinematics of GCs, the velocity distribution of a GCS. Typically this is represented with the standard deviations of the components of velocity in the radial direction (towards or away from the galactic centre) and the tangential direction (motion on the plane perpendicular to the radial velocity) of all GCs at a given radius from the galactic centre, known as the radial and tangential velocity dispersions. Comparison of the radial and tangential velocity dispersions at all radii defines the GCS anisotropy profile. Although there is little in the way of kinematic information on systems of YMCs, the orbit of a cluster will determine the strength of the galaxy tides acting upon it, leading to the preferential destruction of clusters on certain orbits. Thus assuming a reasonable initial anisotropy profile and requiring that

it matches the kinematics of a GCS as observed today (e.g., Pota et al. 2013; Woodley et al. 2010; Lee et al. 2008; Côté 1999; Grillmair et al. 1994; Frenk & White 1980) places another constraint on the evolution of a mass function. A common problem in extragalactic astrophysics is that distances to objects are always far greater than the sizes of the objects themselves. The effect of this is that it is essentially impossible to determine any depth to an observed object — they simply appear as two dimensional objects on the plane of the sky. For example, the only estimate of distance from cluster centre that could be obtained for a star in a cluster would be the perpendicular to line-of-sight, or projected distance, which is always less than or equal to the true distance. The same problem applies when attempting to measure the galactocentric distance of GCs in other galaxies, and to the distribution of light in an observed GC (throughout this thesis, projected distances will always be denoted in upper case, while unprojected distances will be denoted in lower case). The effect of projection with regards to kinematics is that it becomes virtually impossible to resolve individual radial and tangential motions of GCs, such that only projected line-of-sight velocity dispersions may be measured.

Finally there is the GC fundamental plane (GCFP). This is the collective mass-radius (or equivalent) relation for all clusters in a GCS. Theoretical predictions and some data on YMC fundamental plane correlations exist, so a working understanding of how cluster radii change in response to tides and mass-loss allows evolving the GCFP for comparison to what is observed today. However, this requires detailed observations and profile fitting for every GC in a GCS. Of these three GCS observables, the GCLF is by far the easiest to obtain, as all that is required to compile a GCLF is the brightness of GCs in a GCS. Consequently, the GCLF has received the most attention for modelling dynamical evolution of GCSs. Focusing first on properties of individual GCs however, the structural correlations of GCs from McLaughlin & van der Marel (2005) and McLaughlin et al. (2008) are presented in Figure 1.3.

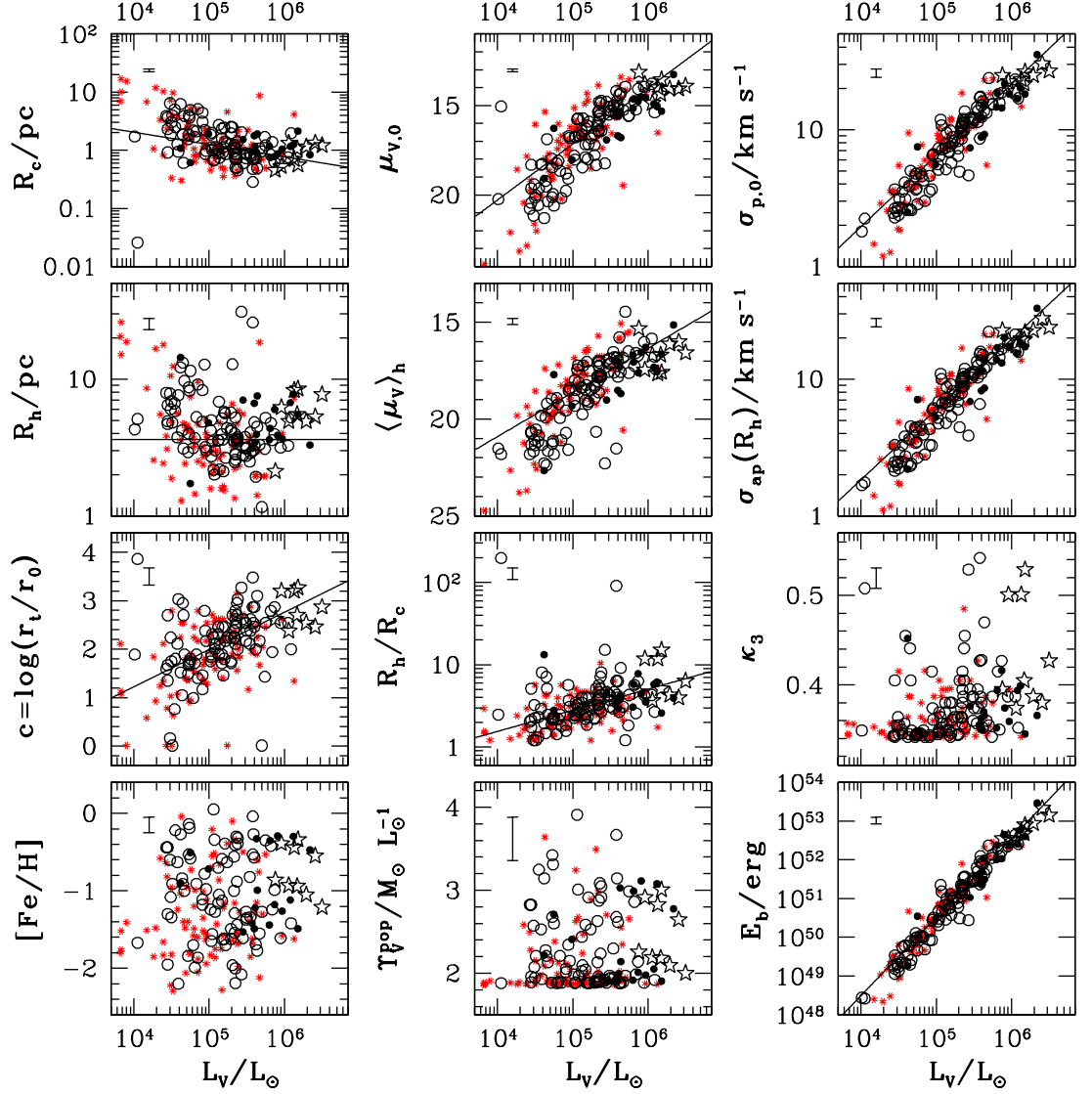


Figure 1.3: Globular Cluster projected core radius (R_c), projected half-light radius (R_h), concentration (c), metallicity ($[\text{Fe}/\text{H}]$), core surface brightness ($\mu_{V,0}$), half-light surface brightness ($\langle \mu_V \rangle_h$), mass-to-light ratio (Υ_V^{pop}), core line-of-sight velocity dispersion ($\sigma_{p,0}$), half-light line-of-sight velocity dispersion ($\sigma_{ap}(R_h)$), $\kappa_3 \equiv 3^{-1/2}(\log[\sigma_{p,0}^2] - \log[\Upsilon_V < \mu_V >_h R_h])$, and binding energy (E_b) as functions of luminosity. Data from the Milky Way (asterisks), large and small Magellanic clouds and Fornax (filled circles; McLaughlin & van der Marel 2005) and from NGC 5128 (open circles and stars; McLaughlin et al. 2008).

As can clearly be seen, there are many interrelations of cluster properties. Note that these panels contain data from several galaxies and yet follow many of the same correlations, indicating that for many GC properties, the host galaxy is unimportant. Where there is mainly scatter, the lack of correlation is visible over data sets from any one galaxy, and over the combined data sets, indicating that the lack of correlation is again not due to different environments for the GCSs, but is somehow intrinsic to GCs. By first making some reasonable assumptions about cluster age and initial conditions, such as initial stellar mass function, initial metallicity, and instantaneous rather than ongoing star formation, given the total luminosity in a certain bandpass (such as e.g., V -band), population synthesis models can predict the mass-to-light ratio, Υ_V^{pop} , which may be used to calculate the mass of a cluster (e.g., Bruzual & Charlot 2003; Maraston 2005). This mass-to-light ratio across entire GCSs is largely consistent around $\Upsilon_V^{pop} \simeq 2M_\odot L_\odot^{-1}$ (McLaughlin & van der Marel 2005), centre bottom panel of Figure 1.3. Once the mass-to-light ratio has been obtained for all GCs in a GCS (either through population synthesis, or assuming $\Upsilon_V^{pop} \simeq 2M_\odot L_\odot^{-1}$), the GCLF may easily be turned into the GCMF. The other properties in Figure 1.3 may then be derived by fitting models to the observed light distribution converted to a mass distribution — a very popular choice is the King (1966) family of models (see section 3.3.2 for a detailed description). The phase space distribution function (the function describing the probability of a particle having a given position and velocity) of King (1966) models is that of a modified or “lowered” isothermal sphere, with finite total mass and finite escape radius, given by:

$$f(E) \propto \begin{cases} \exp [(W_0 - E)/\sigma_0^2] - 1 & \text{for } 0 \leq E \leq W_0 \\ 0 & \text{for } E > W_0 \end{cases} \quad (1.12)$$

where W_0 is the maximum energy a particle can have and still be bound to the system, and is equal to the potential energy at the edge of the system.

Additionally, King (1966) models are ergodic (meaning that the distribution function is assumed to be a function of energy only), and consequently all King (1966) models are isotropic (meaning a spherically symmetric velocity distribution, similar

to a spherically symmetric matter distribution). However, modified models exist that allow velocity anisotropy (Michie & Bodenheimer 1963; Gunn & Griffin 1979; Meylan 1987), with distribution function of the form:

$$f(E, L) \propto \begin{cases} \exp[-L^2/(r_a^2\sigma_0^2)] \{\exp[(W_0 - E)/\sigma_0^2] - 1\} & \text{for } 0 \leq E \leq W_0 \\ 0 & \text{for } E > W_0 \end{cases} \quad (1.13)$$

where r_a is the anisotropy radius, where the transition from isotropy in the core to radial anisotropy in the outer parts occurs. The difference in the fit produced by equation (1.13) as opposed to (1.12) is found to be small, and anisotropy considerations are second order. Thus, there is only a single parameter to determine between the continuum of King (1966) family models, the ‘central potential’, W_0 , or equivalently the concentration, defined as $c = \log[r_t/r_0]$, where r_t is the limiting radius and r_0 is the core radius. Hence, the continuum of models allow for a range of internal structures; $c \leq 0$ implies that $r_t \leq r_0$, i.e. the cluster is essentially all constant density core, and therefore is similar to a homogeneous sphere, and $c \rightarrow \infty \equiv W_0 \rightarrow \infty$ returns to the infinite extent and infinite mass isothermal sphere (cf. equations 1.10 and 1.12). The fact that these models have a finite size imitating tidal truncation without the need for extra parameters concerning the host galaxy keeps them simple, and they generally give good fits besides and thus they are a popular choice.

Bearing the effects of projection in mind, in order to fit the light profile with a King (1966) model, it is necessary to assume something about the shape of the cluster (for a King (1966) model, clusters must be assumed to be spherical), and then to project the model before fitting to the observational data. Once the light profile has been fitted, quantities such as both projected and deprojected half-light and core radius, half-light and core velocity dispersions, etc., can be read off. Doing this, it then becomes clear that all the correlations in Figure 1.3 boil down to a few basic correlations.

1. $\Upsilon \sim \log M^0$ with scatter (centre bottom panel of Figure 1.3) — This is just a stellar population effect — a population with the same metallicity and age has a constant mass-to-light ratio.

2. $\log R_h \sim \log M^0$ with large scatter (first row down, first column of Figure 1.3) — Some claim that the data form a ‘V’ in the parameter space due to a break at $\sim 10^6 M_\odot$ — clusters below this mass are more dynamically old (relaxation times less than a Hubble time); i.e. they have evolved away from the initial mass-radius relation that all stellar systems, including galaxies, start with, while clusters (and stellar systems such as galaxies) above this mass are comparatively dynamically young, with relaxation times greater than a Hubble time, and are thus less evolved and still maintain their initial mass-radius relation (Gieles, Baumgardt & Heggie 2010).

3. $\log \sigma^2 \sim \log M$ with some scatter (first row down, third column of Figure 1.3) — This is a consequence of the virial theorem — $\frac{1}{2}M\sigma^2 \simeq M^2/r_g$, where $r_g \sim r_h$ is the gravitational radius. The scatter probably comes from scatter in M with R_h . This correlation strongly indicates that most clusters are indeed virialised systems.

4. $c \sim \log M^{0.8}$ with large scatter (second row down, first column of Figure 1.3) — Along with the adjacent panel, this is indicating a correlation between cluster structure and mass. How much of this is due to a primordial correlation vs a correlation brought about by evolution is unclear. For example, this could be a signature of the aforementioned initial mass-radius relation in the process of being ‘washed out’ as clusters evolve away from their initial conditions. Alternatively, this could be reflecting the tendency for clusters to become more centrally concentrated (higher c) as they evolve (see section 1.4.5).

With the mass-to-light ratio remaining roughly constant at $\Upsilon \simeq 2M_\odot L_\odot^{-1}$, this means that most scatter in GC properties lies in M vs R_h . Thus GC data may be represented on a plane, commonly referred to as the globular cluster fundamental plane.

1.3.1 Fundamental Plane

Being able to represent GC data in terms of just two independent properties led to the invention of the GC fundamental plane (Djorgovski & Meylan 1994; Djorgovski 1995; Dubath & Grillmair 1997), similar to the fundamental plane for elliptical galaxies (Djorgovski & Davis 1987; Dressler et al. 1987). Consequently, a logical question to follow would be, are GCs essentially only miniature versions of elliptical galaxies? It turns out however, that the scalings between properties are very different, as displayed in Figure 1.4.

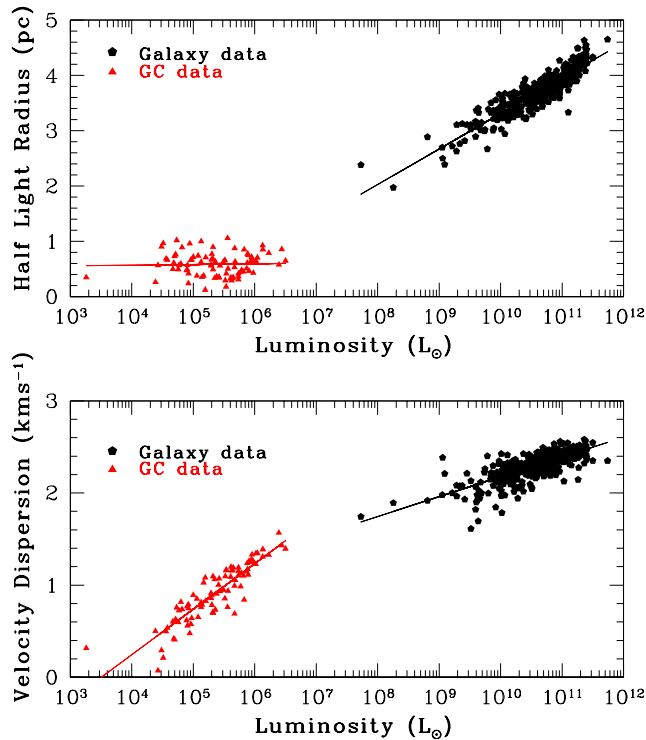


Figure 1.4: Logarithmic projected half-light radii (effective radii for galaxies) and logarithmic velocity dispersions against luminosity for GCs and elliptical galaxies. Elliptical Galaxy data taken from Faber & Gallagher (1979), GC data taken for Milky Way, M31, NGC5128, M33, LMC, SMC, Fornax, taken from McLaughlin & van der Marel (2005).

The top panel clearly shows the lack of a systematic dependence of R_h on L for

GCs, whereas elliptical galaxies have roughly $R_h \propto L^{1/2}$. In the bottom panel, GCs display a $\sigma \propto L^{1/2}$ dependence, whereas elliptical galaxies show a $\sigma \propto L^{1/4}$ dependence. One possible explanation for the differences is that dynamical evolution has re-shaped the GC correlations, but not affected the elliptical galaxy correlations much. Although relaxation time is expected to change with time, the current relaxation time as a fraction of the Hubble time can still be used to indicate how dynamically old a system is; elliptical galaxies tend to have relaxation times greater than a Hubble time, whereas GCs tend to have relaxation times of order a Gyr (e.g., Harris 1996). Additionally, mass-loss due to tidal limitation will tend to accelerate dynamical evolution, and galaxies do not tend to be tidally limited unless they are the minor component of a merger, whereas most GCs are tidally limited. Alternatively, another possibility is that the differences in GC correlations to galaxy correlations may be due to different formation processes. In order to decide between these possibilities, comparable data on the internal properties of YMCs are required. These data are just starting to come into focus, and be used to construct a young-cluster fundamental plane (e.g., Barmby et al. 2009; Bastian et al. 2013). Frequently, YMCs are found to have very distended power-law envelopes, reaching out to several hundred pc (e.g., Elson, Fall & Freeman 1987; Bastian et al. 2013), and to broadly have similar properties from galaxy to galaxy (e.g., Barmby et al. 2009). This could be suggesting that clusters form with a global initial mass-radius relation, but that this gets ‘washed out’ by different rates of dynamical evolution due to different initial masses.

1.3.2 Velocity Distribution

Observing the globular cluster velocity distribution (GCVD) has several difficulties associated with it. Firstly, in order to estimate velocity dispersions for a velocity profile, GCs must be binned radially, and the radial and tangential velocities of GCs in each bin used to calculate the velocity dispersion, e.g. Figure 1.5. However, the Milky Way GCS has only around 150 members, and these span galactocentric distances between about 1 to 100 kpc. Thus either each bin spans a wide range of galactocentric distances,

or there is only a small number of GCs in each bin, causing statistical noise. Despite this, kinematic studies of the Milky Way GCS have been conducted (e.g., Frenk & White 1980; Côté 1999). Frenk & White (1980) fitted models relating rotation of the GCS, spatial distribution of the GCS, and radial and tangential velocity distributions of the GCS to the circular speed of the galaxy. Their best-fitting model required GC

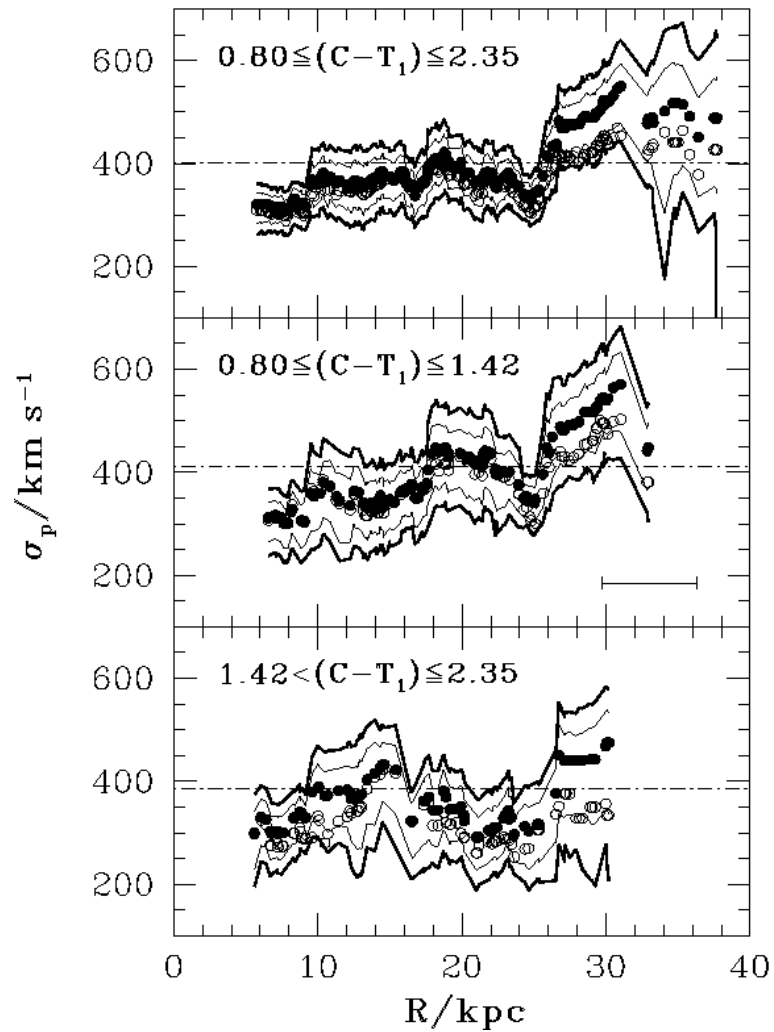


Figure 1.5: Projected line-of-sight velocity dispersions for the GCS of M87 in different colour bands as a function of projected galactocentric distance. The solid curves delimit the 68% and 90% confidence bands, filled circles denote the velocity dispersion in each radial bin, and open circles denote the velocity dispersions corrected for rotation. Taken from Côté et al. (2001).

radial velocity dispersions increasing moderately with galactocentric radius ($\sigma_p \propto r_{gc}^{0.2}$). Côté (1999) compared kinematics of inner GCs ($\lesssim 4\text{kpc}$) with kinematics of atomic hydrogen in the inner Galaxy to support claims that innermost GCs are associated with the central Galactic bulge/bar rather than the thick disk.

Alternatively, the GCVD of a more populous GCS could be studied to overcome statistical difficulties, however this is complicated by projection — only the projected galactocentric distances, which are lower bounds on the true galactocentric distance, can be obtained. Moreover, only the line-of-sight velocities of a cluster can be estimated, which cannot generally be turned into radial and tangential velocities, also due to projection (e.g., Côté et al. (2001) as displayed in Figure 1.5). Thus, the best way to fit a model to a GCS velocity distribution is again to first project the model before fitting to data. The most important detail of Figure 1.5 is that the projected line-of-sight velocity dispersions are roughly constant (i.e. consistent with constant isotropy), or increasing slightly with galactocentric radius (e.g., Frenk & White 1980; G. & Ryzhov 1997; Minniti et al. 1998; Côté et al. 2001; Côté et al. 2003; Bekki et al. 2005; Kafle et al. 2013).

The velocity distribution of a GCS is expected to change over the course of dynamical GCS evolution, as clusters on more elliptical orbits are expected to be destroyed more quickly than clusters on less elliptical orbits (see section 1.4). Thus, the velocity distribution can be used as an additional constraint for GCS evolution models (e.g., Vesperini et al. 2003).

1.3.3 Mass Functions

It has long been established that the GCMF, (number of clusters per logarithmic mass, $dN/d \log M$) in the Milky Way has a distinct peak at about $M_{TO} \sim 1.6 \times 10^5 M_\odot$, and that the peak of the GCMF of other nearby galaxies do not differ from this value significantly. In fact, when a GCS is binned in terms of galactocentric radius, the mass function of clusters in each bin is always the same basic shape with the same peak, i.e. the GCMF is invariant with galactocentric position (e.g., Fall & Zhang 2001; Vesperini

et al. 2003; McLaughlin & Fall 2008). In this light, based purely on empirical evidence the GCMF began to receive attention as a standard candle, with the first attempt to use it to gauge an intergalactic distance by Hanes (1977). Once highly sensitive CCD cameras became available, observations of GCMF peaks of galaxies outside of the Local Group became possible, and the standard candle hypothesis became standard practice (Harris 1991).

With the launch of the HST, the focus quickly shifted to the presence of YMCs in starburst systems. In such systems, the numbers of clusters at low masses continues to rise, as displayed in the top panel of Figure 1.6. The mass function for these young systems are fitted well by a simple power-law, $dN/dM \propto M^{-\beta_{MF}}$, or equivalently $dN/d \log M \propto M^{1-\beta_{MF}}$ with $\beta_{MF} \simeq 2$. Alternatively, the cluster mass functions in starburst systems are also well fitted by a Schechter (1976) function, $dN/dM \propto M^{-\beta_{MF}} \exp(-M/M_{sch})$, which describes a power-law with an exponential cut-off at $M = M_{sch}$, with $M_{sch} \sim 10^6 M_{\odot}$. Essentially, the mass function of these systems are unpeaked (Schweizer et al. 1996; Zhang & Fall 1999; Fall, Chandar & Whitmore 2005; Bastian et al. 2006). On the other hand, the GCMF is better described by a Gaussian in $\log M_c$ (a log-normal distribution), as displayed in the bottom panel of Figure 1.6. Before long, it was noticed that at masses above the GCMF peak mass, globular cluster and YMC mass functions may both be fitted well with a power-law or Schechter (1976) function. Thus if there was some process that could preferentially destroy low-mass clusters, a YMC mass function might be made to resemble a GCMF as observed today, i.e. GCMFs may have originally been well described by a power-law or Schechter (1976) function. Furthermore, this destructive process would have to act in such a way that the GCMF of radially binned GCs is always the same; i.e. it would have to reproduce the GCMF invariance with galactocentric radius.

The question now, as suggested above, is can the unpeaked mass function of YMCs evolve into a mass function peaked at a “characteristic” $M_{TO} \sim 10^5 M_{\odot}$, as observed for old GC systems, given a Hubble time of evolution? And if so, which destruction process is responsible? Several evolutionary processes are well known that might explain such a transition, by eroding clusters such that after a Hubble time, the

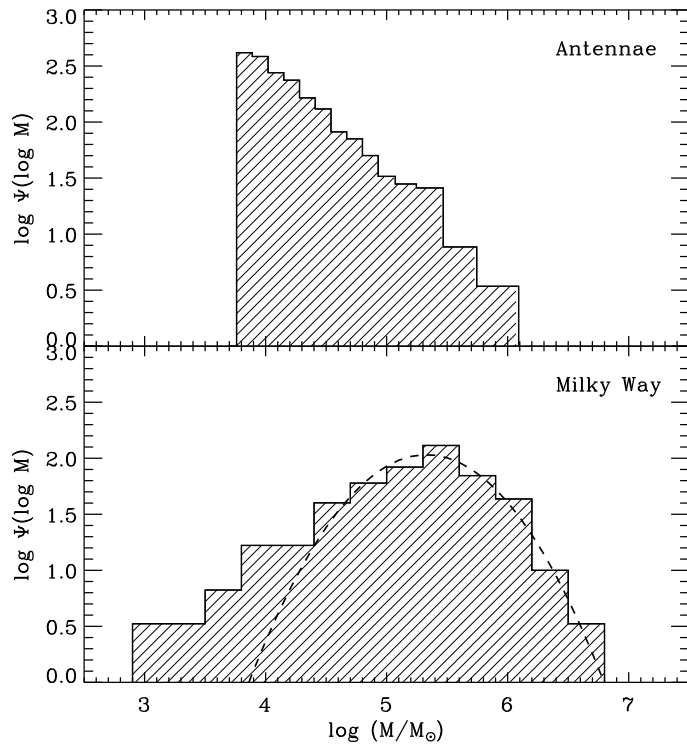


Figure 1.6: The Antennae (top) and Milky Way (bottom) cluster mass functions (Fall & Zhang 2001), where Ψ is the number of clusters with masses between M and $M+dM$ at time t .

mass function resembles GCMFs as observed today (see section 1.4). These evolutionary processes would have to preferentially erode the low-mass end of the GCMF, whilst leaving the more massive end relatively untouched, such that $dN/dM \propto M^{-\beta_{MF}}$ still for $M \gtrsim M_{TO}$ (cf. top and bottom panels of Figure 1.6), resulting in a distinct peak emerging.

That these evolutionary processes occur is well established, but specifically which process is most important is an ongoing question. Some arguments suggest that these processes will be effectively instantaneous and happen early on in the clusters lifetime, imprinting the mass function with the shape as seen today at an early age, with the

other evolutionary processes having little effect to the present day (e.g., Vesperini & Zepf 2003; Parmentier & Gilmore 2007; Baumgardt, Kroupa & Parmentier 2008). Other arguments contend that these instantaneous processes will certainly occur, but will deplete the same fraction of mass in all clusters such that the shape of the GCMF remains unchanged, unless special conditions are invoked. The actual change of shape will occur slowly over Gyr time-scales due to other processes (e.g., Fall & Rees 1977; Caputo & Castellani 1984; Fall & Zhang 2001; Jordán et al. 2007; McLaughlin & Fall 2008). The possible destruction mechanisms and their effect on the GCMF will be discussed in section 1.4, followed by a discussion of which is the most likely candidate for turning an initial power-law CIMF into a GCMF with a distinct peak.

1.4 Cluster Destruction Mechanisms

1.4.1 Stellar Evolution

The effects of stellar evolution on clusters can be broken down into two basic regimes: mass-loss caused by massive stars early on in the host cluster life, and the mass-loss associated with stars expiring and becoming degenerate remnants (which may subsequently be ejected from the cluster by a ‘kick’ velocity).

Given a system of total mass M_0 and radius r_g that reaches virial equilibrium before any significant mass-loss, then the velocity dispersion is given by the virial theorem to be $\sigma^2 = GM_0/r_g$. Then for mass-loss that occurs on a time-scale shorter than the dynamical time-scale of the cluster, the velocity dispersion will remain unaffected. If the spatial distribution of the system is also unaffected, then the energy of the remaining system is given by $E = 0.5M\sigma^2 - GM^2/r_g$, which once virial equilibrium is reached again will also be given by $E = -GM^2/2R$, where R is the new radius of the system, and the mass-loss is given by $\Delta M = M_0 - M$. In this case, the final radius is related to the initial radius by $R/r_g = (M_0 - \Delta M)/2(M_0/2 - \Delta M)$ (Hills 1980). Then if the mass loss $\Delta M \geq 0.5M$ in less than a dynamical time, the system will dissociate. This

scenario adequately describes clusters that are still embedded in their pre-natal clouds — a significant portion of mass is in gas, with the rest in stars. The most massive stars have strong winds and short lifetimes that end with supernovae explosions, making them adept at rapidly expelling gas out of the cluster, whilst not having much of an effect on other stars in the cluster. If a young cluster has not reached virial equilibrium before massive stars start causing significant mass-loss, even less mass-loss is required to dissociate the cluster (Hills 1980).

With an initial stellar mass function providing few massive stars to begin with, if a cluster survives this early phase (the first 100 Myr or so), then there will be very few massive stars remaining such that mass-loss from the cluster due to stellar winds may be neglected, and remaining mass-loss by stellar evolution is primarily in the form of stars evolving off of the main-sequence and turning into stellar remnants. Then ignoring changing stellar mass due to winds, and assuming that a star instantly turns into a remnant upon evolving off of the main-sequence, the mass of a main-sequence progenitor expiring at time t is denoted as $m_{to}(t)$. Remnant mass as a function of progenitor mass is given by $m_{rm}(m)$, and the probability of an asymmetric explosion upon expiry imparting a sufficient ‘kick’ or recoil velocity to the remnant to eject it from the cluster is $P_{ej}(m)$. Then upon the expiry of all stars of mass m , the cluster will lose mass

$$dM_{loss}(m) = \frac{dN}{dm} [m - \{1 - P_{ej}(m)\}m_{rm}(m)] dm$$

where $\frac{dN}{dm}$ is the initial stellar mass function defined between the most and least massive stars present, m_u and m_l . Then the total mass remaining by time t is given by:

$$M(t) = M_{init} - \int_{m_{to}(t)}^{m_u} \frac{dN}{dm} [m - \{1 - P_{ej}(m)\}m_{rm}(m)] dm \quad (1.14)$$

with

$$M_{init} = \int_{m_l}^{m_u} \frac{dN}{dm} m dm \quad (1.15)$$

The functions $P_{ej}(m)$ and $m_{rm}(m)$ are provided by stellar evolution theory (with additional constraints placed by observations), (e.g., Iben & Renzini 1983; Hurley, Pols & Tout 2000; Heyl 2007; Woosley & Heger 2007), though these are broken down into different classes of remnant, namely white dwarfs, neutron stars, and black holes, depending on the progenitor mass. $P_{ej}(m)$ is likely to be quite low for white dwarfs, as the end phase of a star turning into a white dwarf is not particularly violent, whereas $P_{ej}(m)$ for neutron stars and black holes is likely to be much higher due to the violent supernova explosion. Additionally, the initial stellar mass function can be obtained by observations (e.g., Kroupa 2001; Chabrier 2003). As the initial stellar mass function, dN/dm , the ejection probability, $P_{ej}(m)$, and the remnant mass as a function of progenitor mass, $m_{rm}(m)$, will be the same for all clusters, and dN/dm is normalised to the mass of each cluster (cf. equation 1.15), mass-loss due to stellar evolution at time t will be the same fraction of initial cluster mass for all clusters. Consequently, mass-loss due to stellar evolution only rescales a GCMF, and does not change its shape, ruling it out as a candidate for turning an initial power-law CIMF into a GCMF as observed today over 13 Gyr.

1.4.2 Dynamical Friction

Dynamical friction is an effect caused by a continuum of background particles that leech orbital energy and orbital angular momentum from a subject body via gravitational interactions. As the subject body passes through a region of space, it perturbs field particles such that a ‘gravitational wake’ is formed. The gravitational effect of this wake then serves to decelerate the subject body, decreasing both its orbital energy and orbital angular momentum, putting it on a new orbit from one moment to the next. Since orbital energy and orbital angular momentum are sapped at different rates that vary around an orbit depending on the subject body speed and orbital angular momentum, ellipticity is not necessarily conserved. The average of these changes over a radial period will determine whether a subject body is being placed on either a more or less elliptical orbit overall (see section 5.1). Thus dynamical friction may play a role in

evolving the velocity anisotropy profile of GCSs. Furthermore, dynamical friction will serve to bring clusters in closer to the galactic centre where galactic tides are stronger, and hence may hasten the destruction of some clusters. As a result, dynamical friction will also play a role in evolving the GCMF (e.g., Vesperini 2000).

Under the assumptions that the subject body is much more massive than field stars but much less massive than the host system, that the host system is sufficiently larger than the subject body as to be considered infinite and homogeneous, and that the distribution function describing the field stars is Maxwellian, Chandrasekhar (1943) calculated the deceleration of a subject body due to dynamical friction, obtaining (e.g., Binney & Tremaine 2008):

$$\begin{aligned}\frac{d\mathbf{V}}{dt} &= -\frac{4\pi G^2 M_c \rho_f \ln \Lambda}{V^3} \left[\text{erf}(X) - \frac{2X}{\sqrt{\pi}} \exp(-X^2) \right] \mathbf{V} \\ X &= V/\sqrt{2\sigma^2}\end{aligned}\tag{1.16}$$

where σ is the velocity dispersion of the field stars, r_M is the galactocentric distance of the subject body, \mathbf{V} is the velocity of the subject body with V its speed, erf is the Error function, $\ln \Lambda$ is the Coulomb logarithm as in section 1.2.2, with $\Lambda = b_{max}/b_{min}$, where b_{max} and b_{min} are the maximum and minimum impact parameters of the predominant long range encounters. Typically, $b_{max} \sim r_M$ and $b_{min} \sim \max(r_h, GM_c/\sigma^2)$ where r_h is the radius containing half the mass of the subject body (see section 5.2.4 for a more refined evaluation of Λ).

For circular orbits, the vector notation may be discarded. Also assuming a singular isothermal sphere such that $X = 1$ and $\rho_f(r) = \frac{V^2}{4\pi Gr^2}$, this deceleration multiplied by the radius of the cluster orbit is equal to the torque acting on the subject body, such that:

$$\frac{dL}{dt} \simeq \frac{1.71\pi G^2 M_c^2 \rho_f r_M \ln \Lambda}{V^2}\tag{1.17}$$

As the rotation curve of an isothermal sphere is flat, the orbital speed, V , will remain constant as the cluster spirals inwards, such that $L = M_c r_M V$ with V a constant.

This can be substituted into (1.17), and the differential equation solved to achieve an expression for the time required for a cluster to spiral to the centre of the galaxy (Binney & Tremaine 2008):

$$t_{df} \simeq \frac{19\text{Gyr}}{\ln \Lambda} \left(\frac{r_{M,i}}{5\text{kpc}} \right)^2 \frac{\sigma}{200\text{km s}^{-1}} \frac{10^8 M_\odot}{M_c} \quad (1.18)$$

where $r_{M,i}$ is the initial galactocentric distance. Clearly, the time-scale for this process is very long for all but the most massive or the innermost clusters. Thus, the destruction rate due to dynamical friction is given by:

$$\mu_{df} = \frac{GM \ln \Lambda}{1.65 r_{M,i}^2 \sigma} \quad (1.19)$$

In terms of mass, dynamical friction will only have a noticeable effect on the high-mass end of the GCMF, and thus is not a contender for turning initial power-law CIMFs into GCMFs as observed today. For a more detailed treatment of dynamical friction affecting cluster orbits, see section 5.1.

1.4.3 Tidal Shocks

A tidal shock is where the local potential experienced by the cluster varies on a time-scale shorter than the relaxation time of the cluster. This can happen when a cluster passes close to or through an area of high-density, such as the central regions for spherical galaxies, or the bulge or disk for disk galaxies. Following Spitzer (1987), supposing a galaxy disk that is infinite and thin, the acceleration on opposite sides of the disk is $2\pi G\Sigma$, where Σ is the surface density of the disk. As a cluster crosses the disk, there will be a point when one of the cluster's stars and the cluster centre-of-mass will be on opposite sides of the disk. The star is then accelerated relative to the cluster centre-of-mass by $4\pi G\Sigma$, and this acceleration will last for a time $r_z/v_{M,z}$ where r_z is the perpendicular component of the displacement of the star to the cluster centre-of-mass, and $v_{M,z}$ is the perpendicular component of the cluster velocity relative to the disk.

Assuming that the duration of the shock is much shorter than a cluster crossing time (the impulse approximation), so that for the duration of the encounter, the cluster stars do not change position, the star has its speed altered by $\Delta v_s = 4\pi G \Sigma r_z / v_{M,z}$. The cluster then gains energy $\Delta E \sim 0.5 M_c (\Delta v_s)^2 \sim 8\pi^2 G^2 M_c \Sigma^2 r_z^2 / v_{M,z}^2$. Once the cluster has experienced enough shocks such that the injected energy becomes comparable to the cluster binding energy, $\Delta E \sim E$, the cluster will become unbound. Writing the cluster internal energy as $E \sim G M_c^2 / r_g$, then since a cluster will suffer two disk crossings per azimuthal orbit, the time to disruption by shocking is given by $t_{sh} \sim 0.5 T_\psi E / \Delta E \sim T_\psi M_c v_{M,z}^2 / 4\pi^2 G \Sigma^2 r_z^3$, where T_ψ is the cluster azimuthal period and $r_z \sim r_g$ has been assumed. Thus the time to disruption due to disk shocking is given by (e.g., Heggie & Hut 2003; Binney & Tremaine 2008):

$$t_{sh} \simeq 417 \text{ Gyr} \frac{T_\psi}{200 \text{ Myr}} \left(\frac{v_{M,z}}{130 \text{ km s}^{-1}} \right)^2 \left(\frac{50 \text{ M}_\odot \text{ pc}^{-2}}{\Sigma} \right)^2 \frac{\rho_h}{50 \text{ M}_\odot \text{ pc}^{-3}} \quad (1.20)$$

where ρ_h is the cluster density. Disk galaxy bulge and spherical galaxy core shocking are very similar processes to one another, but differ from disk shocking in several ways. For example, the duration of a bulge shock is not necessarily shorter than a typical cluster crossing time, and thus the impulse approximation breaks down. By treating the cluster as a harmonic potential (assuming constant density), Gnedin & Ostriker (1997) derive for the time to disruption due to bulge shocking:

$$t_{sh} \simeq 367 \text{ Gyr} \frac{T_r}{500 \text{ Myr}} \left(\frac{v_p}{500 \text{ km s}^{-1}} \right)^2 \left(\frac{r_p}{0.5 \text{ kpc}} \right)^4 \left(\frac{10^9 \text{ M}_\odot}{M_b} \right)^2 \frac{\rho_h}{50 \text{ M}_\odot \text{ pc}^{-3}} \quad (1.21)$$

where v_p and r_p are the cluster speed and galactocentric distance at orbital pericentre, T_r is the cluster radial period, and M_b is the host galaxy bulge mass.

Shocking is therefore most effective at destroying clusters with low densities, and with small orbits such that clusters suffer more frequent disk crossings/bulge passages at higher speeds, in addition to higher disk surface density or smaller pericentres. The mass-loss rate is given by $\mu_{sh} = M_c / t_{sh}$. Using equation (1.21), this is:

$$\mu_{sh} = \frac{7G}{9\pi} \left(\frac{M_b}{v_p r_p^2} \right)^2 \frac{M_c}{T_r \rho_h} \quad (1.22)$$

Thus, the destruction rate for clusters scales as $\mu_{sh} \propto M_c/\rho \propto r_{char}^3$, where r_{char} is some characteristic radius. Gnedin, Lee & Ostriker (1999) found that as clusters lose mass, this characteristic radius shrinks, such that mass-loss due to shocks is self limiting. As a result, tidal shocking is expected to dominate during the intermediate stages of a cluster's lifetime, after stellar evolution has run its course. However, tidal shocking will eventually become weak, and other mass-loss mechanisms will dominate instead, (e.g., Fall & Zhang 2001). Moreover, the destruction rate due to tidal shocks is very dependent on galactocentric radius, through either the azimuthal or radial period, and either disk surface density or pericentric distance for disk and bulge shocks respectively. Thus tidal shocks would be unlikely to reproduce the GCMF invariance with galactocentric position while whittling a power-law CIMF into a GCMF as observed today.

1.4.4 Evaporation

Evaporation is the loss of stars that have achieved escape velocity via two-body interactions, and thus is closely linked to relaxation (section 1.2.2). In fact, the rate that stars achieve escape energy occurs on the relaxation time-scale, which is the same whether a cluster is isolated or tidally limited (though escapers from isolated systems result mainly from single energetic encounters, rather than the culmination of many weak interactions as is the case for tidally limited systems e.g., Spitzer 1987). As stars escape, they carry away energy in the form of kinetic energy and each escaper's contribution to the cluster gravitational potential. This process of relaxation driven evaporation will continue until mass and energy loss of the system are sufficiently severe as to cause it to become unbound. Submerging this system in the gravitational potential of a host galaxy imposes a tidal limit such that particles only need sufficient energy to reach the tidal radius rather than infinity to escape, and thus evaporation is

much swifter. Dimensionally then, evaporation rate scales as the total mass over the relaxation time, and will have a different numerical coefficient depending on whether a cluster is isolated or tidally limited (Spitzer 1987). For an isolated cluster, the squared escape speed is given by $v_{esc}^2(r) = -2\Phi(r)$, and so the mean-square escape speed is given by $\langle v_{esc}^2 \rangle = \int_0^\infty \rho(r) v_{esc}^2(r) d^3r / \int_0^\infty \rho(r) d^3r = -4W/M$. According to the virial theorem, $-W = 2K = M \langle v^2 \rangle$, and hence:

$$\langle v_{esc}^2 \rangle = 4 \langle v^2 \rangle \quad (1.23)$$

writing the velocity distribution function as $f(v)$, then:

$$\langle v^2 \rangle = \frac{\int_0^\infty v^2 f(v) d^3v}{\int_0^\infty f(v) d^3v}$$

Hence, the fraction of the velocity distribution expected to escape is given by:

$$\xi = \frac{\int_{\langle v_{esc}^2 \rangle^{1/2}}^\infty f(v) d^3v}{\int_0^\infty f(v) d^3v}$$

Once these particles with sufficient energy to escape have done so, the velocity distribution will lack a high-velocity tail. Thus the cluster stars will begin to redistribute their energies so as to return to the original velocity distribution on a relaxation time-scale (see section 1.2.2). Hence, ξ is the fraction of stars (and therefore cluster mass) lost per relaxation time (Ambartsumian 1938).

For a Maxwellian speed distribution, $f(v) \propto v^2 \exp(-v^2/\sigma^2)$, this is $\xi = 0.00738$ (e.g., Spitzer & Härm 1958). However, in a tidally limited cluster, escaping stars need only reach the tidal boundary rather than infinity, and thus the mean-square escape speed is reduced by $2GM/r_t$ (e.g., Spitzer 1987), giving $\langle v_{esc}^2 \rangle = 4 \langle v^2 \rangle \left(1 - \frac{GM}{2\langle v^2 \rangle r_t}\right)$. Combining this with $r_h \simeq 0.45 \frac{GM}{\langle v^2 \rangle}$ (see section 1.2.1), the following is obtained:

$$\langle v_{esc}^2 \rangle \simeq 4 \langle v^2 \rangle \left(1 - \frac{10}{9} \frac{r_h}{r_t}\right) \quad (1.24)$$

This additional dependence on r_h/r_t signifies mass-loss rate dependence on cluster structure. Figure 1.7 displays r_h/r_t as a function of concentration c for King (1966)

models, showing that for these models $r_h/r_t \in (0.1, 0.35)$ for a range of c including most Milky Way GCs. One subtlety of this line of thought however is that the implicit assumption that r_t means the tidal radius has been made, i.e. that the cluster in question is in fact filling its Roche lobe. Alternatively, r_t could mean the zero density radius which is in fact inside of the tidal radius. For a very compact cluster, $r_h \ll r_t$ and the escape speed in equation (1.24) returns to that of an isolated cluster in equation (1.23). Thus tidal limitation lowers the mean-square escape speed, resulting in a greater fraction of the speed distribution escaping per relaxation time.

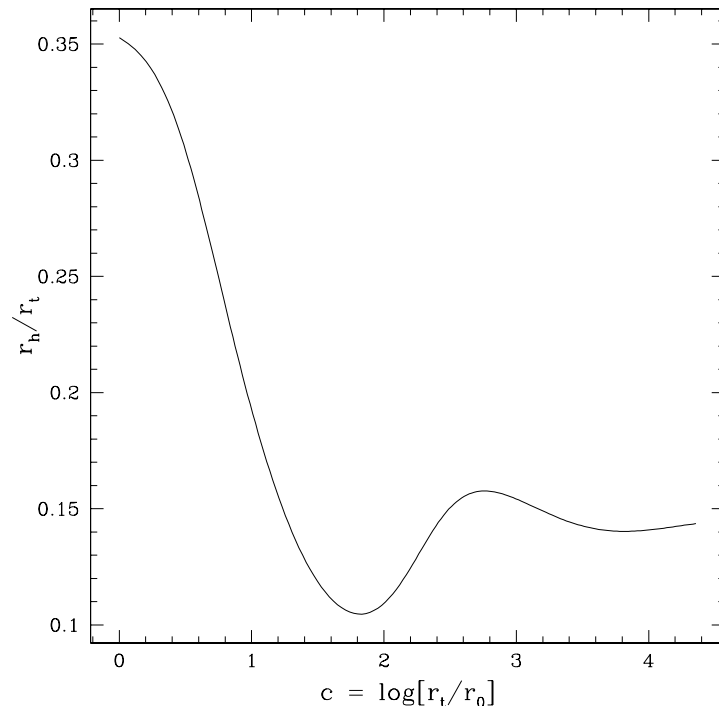


Figure 1.7: The ratio of half-mass radius, r_h , to tidal radius, r_t , as a function of cluster concentration, $c = \log[r_t/r_0]$, for King (1966) models.

Dimensionally then, the rate of mass-loss due to evaporation scales as $\mu_{ev} \propto \xi M_c/t_{rh} \propto \xi \rho_h^{1/2}$ (ignoring the weak dependence on M_c in the Coulomb logarithm). However, given the dependence of ξ on r_h/r_t , the rate of destruction by evapo-

ration is more accurately written as $\mu_{ev} \propto M_c f(\rho_t/\rho_h)/t_{rh}$, where f should tend to ξ as calculated for isolated clusters when Roche lobe under-filling, and to ξ as calculated for tidally limited clusters when Roche lobe filling. Unfortunately, even $\mu_{ev} \propto M_c f(\rho_t/\rho_h)/t_{rh}$ is still an oversimplification, as this is implicitly assuming that stars leave the cluster as soon as they reach escape velocity (the energy criterion). In reality, a star becomes a potential escaper and must first reach and cross the tidal radius before actually escaping (the apocentre criterion, Fukushige & Heggie 2000). Whilst travelling to the tidal radius, this potential escaper may have a close encounter with another star, causing it to lose energy such that escape is no longer possible (King 1959). Since the time-scale for motion of stars inside a cluster is the crossing time, t_c , it seems logical to expect the time-scale for escape when considering this effect to be a combination of the relaxation time and the crossing time. Baumgardt (2001) investigated mass-loss under the apocentre criterion, and found that the time-scale for escape was well represented by the combination $t_{ev} = t_{rh}^x t_c^{1-x} \propto M_c^x \rho_h^{-1/2}$. Also bearing in mind that as a cluster on an eccentric orbit moves away from pericentre, the growing tidal radius may envelop some of the recently lost stars, and they may become bound to the cluster again (see section 1.2.3). Thus the time averaged mass-loss of a cluster around an orbit will not correspond to the tides at pericentre, but will lie further away at some radius r_{ev} . Thus the mass-loss due to evaporation is generally given by:

$$\mu_{ev} \propto \xi M_c^{1-x} \rho_h^{1/2} f\left(\frac{\rho_t(r_{ev})}{\rho_h}\right) \quad (1.25)$$

The main uncertainties in evaporation rate then, are the value of $0 \leq x \leq 1$ determining how much evaporation is slowed by escape rate dependence on crossing time, the galactocentric radius corresponding to the tidal density that matches time averaged mass-loss, r_{ev} , and the functional form of $f(\rho_t/\rho_h)$ representing the escape speed dependence on r_h/r_t , i.e. depending on cluster internal structure/degree of Roche lobe filling.

Baumgardt (2001) first explained the evaporation time-scale scaling with a combination of relaxation time and crossing time, and obtained an estimate of $x \simeq 0.75$.

Additionally, Baumgardt (2001) found that despite the additional dependence on M_c , the mass-loss rate is still approximately constant over a cluster lifetime. This was largely corroborated by the findings of Tanikawa & Fukushige (2005) and Tanikawa & Fukushige (2010). They also found that x depends on the density distribution of the host galaxy and on internal cluster structure. However, since they did not account for mass-loss rate dependence on r_{ev} or $f(\rho_t/\rho_h)$, this could actually just be due to degeneracy between x , r_{ev} and $f(\rho_t/\rho_h)$.

Several studies have been conducted on exactly what r_{ev} should be, with some saying it should be the orbital pericentric radius (von Hoerner 1957), others claiming it should be the orbital radius of the time averaged tides (Küpper et al. 2010), and others claiming it should be some function of eccentricity multiplying the orbital pericentric radius (Webb et al. 2013). There is no clear resolution to these different assertions as yet.

The dependence of mass-loss rate on (ρ_t/ρ_h) has been treated in several ways, such as assuming a homologous cluster structure (Hénon 1961) meaning that at any time, cluster structure is just a rescaled version of cluster structure at any other time, i.e. $\rho_t/\rho_h = \text{const}$, such that $r_h/r_t = 0.145$ and thus $f = \text{const}$ at all times as the cluster loses mass. Baumgardt (1998) assumed a functional form of $f = \sqrt{1 + (\alpha r_h/r_t)}$ with α determined from the results of N-body simulations. Alternatively, Gieles & Baumgardt (2008) arrived at a functional form of $f = \xi_0 \exp(10r_h/r_t)$ where ξ_0 is the fraction of mass lost per relaxation time for a cluster in isolation (e.g., Spitzer 1987). A more thorough review of research into evaporation rates will be given in section 5.2.1.

Rather than identifying and explaining each of these uncertainties individually, other studies have focussed instead on utilising only more easily observed GC data, i.e. their half-light densities and luminosities, and employing assumptions about the initial GCMF. Equation (1.25) is then employed with all of the uncertain terms bunched into a constant, and the GCMF is evolved through time. The constant of combined uncertainties that gives the closest match to observed GCMFs today then gives a numerical estimate of the product and spread of these uncertainties in a GCMF, and may be compared to theoretical results and simulations. Generally, the agreement

is quite good (e.g., Jordán et al. 2007, McLaughlin & Fall 2008, Chandar, Fall & McLaughlin 2007, Chapter 2). Treating the uncertainties in equation (1.25) in this way returns the mass-loss rate to the much simpler relation of $\mu_{ev} \propto \rho_h^{1/2}$. See Chapter 2 for a more detailed description of this process.

In terms of evolving a power-law CIMF into a log-normal GCMF, evaporation is a strong candidate. This is because for $x \simeq 1$ (e.g., Baumgardt 2001; Tanikawa & Fukushige 2005; Tanikawa & Fukushige 2010; Lamers, Baumgardt & Gieles 2010) evaporation depends only weakly on cluster mass. Assuming that King (1966) models are an accurate description of real GCs, then given that r_h/r_t does not vary much (a factor of about 3 over realistic cluster concentrations, cf. Figure 1.7), then the function $f(\rho_t/\rho_h)$ is also likely to not vary very much. Therefore, the mass-loss rate is likely to be roughly proportional to half-mass density. While the expectation may be that tidal limitation will impose a correlation between half-mass density and galactocentric position, plots of ρ_h vs r_M do have some visible trend, but is largely scatter as displayed in Figure 1.8. Therefore, μ_{ev} is only very weakly dependent on galactocentric position, which is consistent with GCMF radial invariance. Moreover, when a GCS is binned in terms of GC internal density as in Figure 1.9, the GCMF of higher density bins is found to shift to higher masses, in a way that $\mu_{ev} \propto \rho_h^{1/2}$ naturally explains. If tidal shocking were the dominant destruction mechanism ($\mu_{ev} \propto M_c/\rho_h$), the opposite would be expected, i.e. higher density bins would be expected to have undergone less mass-loss, and therefore be at lower masses than lower density bins. If dynamical friction were the dominant destruction mechanism ($\mu_{ev} \propto M_c$), the GCMF would be expected to be independent of internal cluster density. Thus evaporation seems to be the dominant destruction mechanism driving GCMF evolution.

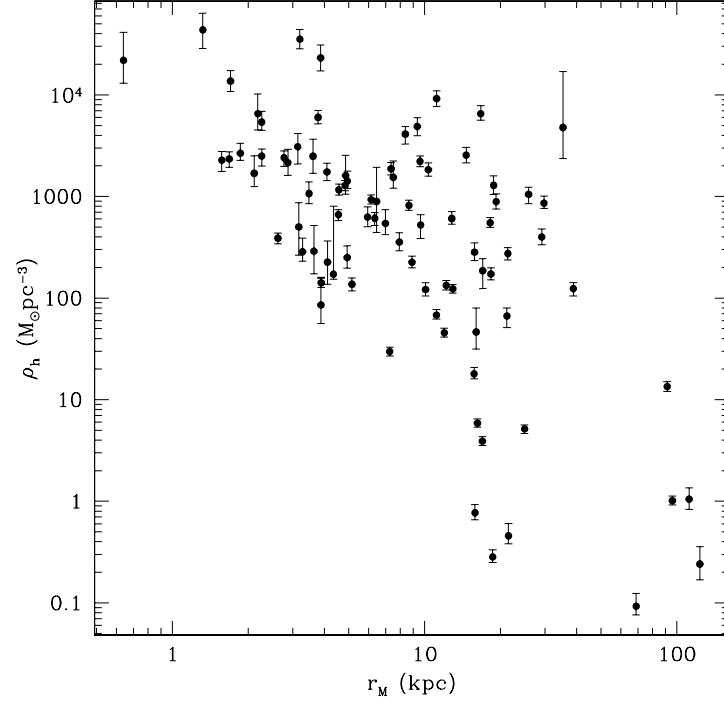


Figure 1.8: Internal GC half-mass density vs galactocentric position for the Milky Way GCS.

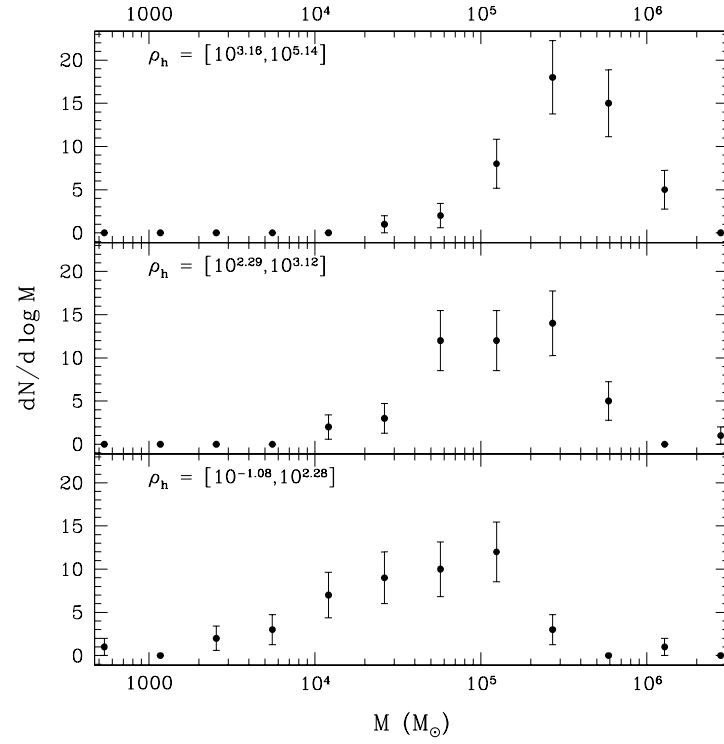


Figure 1.9: Mass functions of Milky Way GCs of different internal density ranges. The density range in each bin is indicated in the top left of each panel (in units of $M_{\odot}\text{pc}^{-3}$).

1.4.5 Core Collapse

Relaxation serves to redistribute energy throughout a cluster, with faster stars generally donating kinetic energy to slower stars of the same mass during an encounter. Stars that gain energy will move onto larger orbits and consequently pass closer to the tidal boundary, resulting in their preferential evaporation. Those that lose kinetic energy will sink lower into the cluster potential, accelerating as they do so and consequently will tend to have more kinetic energy than other stars they encounter, and thus will tend to transfer more kinetic energy away. As relaxation time depends inversely on density, transfer of kinetic energy is more rapid in the central regions. Consequently, sinking stars will rapidly exchange kinetic energy with local stars, which will then carry that energy into the outer envelope. As density is much lower in the outer envelope, relaxation is inefficient at bringing that energy back into the core. Thus sinking stars continue to have more kinetic energy than other stars around them, and consequently continue to transfer away more kinetic energy and sink further into the cluster potential. This process is called gravothermal instability, and serves to dramatically increase cluster concentrations (Lynden-Bell & Wood 1968), meaning the core becomes relatively much more dense while the halo becomes relatively much more diffuse. Were this process to continue unhindered, the cluster would form a singular (infinite density) core in finite time (e.g., Spitzer & Thuan 1972), hence the name core collapse.

When a distribution of stellar masses is present, the onset of core collapse is even more rapid. This is because encounters tend to equalise kinetic energies, so massive stars tend to donate energy to less massive stars even if they have a lower speed. The result of this is that the average kinetic energies of different local mass classes will tend to equalise, $m_1 \langle v_1^2 \rangle = m_2 \langle v_2^2 \rangle$, known as energy equipartition, where $m_1 \gg m_2$ are masses of two stellar populations. The more massive population will tend to sink while the less massive population will migrate outwards due to the energy changes, where they will then begin to move towards energy equipartition in their new local regions. Ultimately this leads to mass segregation, where the positions and masses of cluster

stars are correlated. This has the further effect of placing the most easily accelerated (least massive) stars nearest to the tidal boundary, and more low-mass stars achieve escape velocity (equation 1.24) such that a mass segregated cluster preferentially loses its least massive stars.

As massive stars sink into the cluster potential, they also accelerate, such that their mean kinetic energy is likely to exceed that of stars at the new more central position. If the total mass in the more massive stars, M_1 , is greater than the total mass in the local less massive stars, M_2 , then they will form an independent self gravitating subsystem at the core of the less massive stars. Kinetic energy will continue to be transferred to any less massive stars that stray close enough, and the subsystem of massive stars will continue to contract, accelerating in the cluster potential as they do so, such that they evolve away from, rather than towards, equipartition. This process, known as the equipartition instability (Spitzer 1969), will continue indefinitely if $M_1 \gtrsim M_2(m_2/m_1)^{3/2}$ (e.g., Heggie & Hut 2003; Binney & Tremaine 2008), leading ultimately to core collapse.

If core collapse were to continue unabated, the core densities of clusters would reach infinite values within a finite time, possibly leading to a massive black hole (Marchant & Shapiro 1980) or some other exotic physics such as stellar coalescence (Lee 1987). However, core collapse is probably not actually this catastrophic, and will eventually be halted by binary systems in the core before stellar collisions become likely (Hut 1986). Binaries with less binding energy than the mean kinetic energy of cluster stars are called soft binaries, and tend to dissociate with close encounters with other stars. Binaries with more binding energy than the mean kinetic energy of cluster stars are called hard binaries, and close encounters will cause the binary to become more tightly bound whilst boosting the kinetic energy of the third star (Heggie 1975; Hills 1975). Since when considering long-range interactions, close binary pairs can be treated as a single massive particle, relaxation will tend to cause primordial binaries to sink into the core (e.g., Koch et al. 2004). Due to the greater likelihood of hard binaries surviving encounters, most of these core binaries will be hard binaries. In the absence of primordial hard binaries in the core, as the core undergoes collapses core density will

increase by many orders of magnitude to the point that typical encounters between stars become close enough to lead to captures, resulting in the formation of binary pairs (cf. section 1.2.2). Thus either with primordial binaries or not, hard binaries are expected to be present in the cores of GCs providing a source of energy by boosting the kinetic energies of the population of sinking stars, halting the collapse (Heggie 1979). The rate at which binaries re-energise the core is strongly dependent on core density, which leads to oscillations between core collapse and binary re-heating, known as gravothermal oscillations (Sugimoto & Bettwieser 1983; Goodman 1987; Makino 1996).

Evolution of clusters after the first 100Myr or so, once all the supernovae have run their course, is expected to depend upon initial conditions (e.g. some kind of initial mass-radius relation). This evolution is governed by relaxation, and will eventually lead to core collapse after some time depending on the aforementioned cluster initial conditions. These initial conditions will be erased when core collapse reconfigures cluster internal structure, leaving all post core collapse clusters on similar evolutionary tracts. Consequently, subsequent evolution is just a function of remaining mass and tides (Küpper, Kroupa & Baumgardt 2008; Hurley & Mackey 2010). Around 20% of Galactic GCs are thought to have undergone core collapse already (Harris 1996).

The impact of core collapse on mass-loss is that post core collapse clusters are destroyed more efficiently (Chernoff & Djorgovski 1989; Lee & Goodman 1995). This is because highly concentrated clusters have much bigger halos than cores (core oscillations involve only the inner 1-2% of cluster mass; Goodman 1987) and consequently most mass is in the halo where it is more vulnerable to tidal shocking.

1.4.6 Destruction Mechanism Overview

As stellar evolution is responsible for less mass-loss as time elapses, and is expected to remove the same fraction of mass from all clusters (assuming they all start with a similar stellar mass function), the focus in this overview is placed on the slow destruction mechanisms whose ongoing effects can be studied today. As dependencies on

galactocentric position manifest through factors such as host galaxy volume density, host galaxy surface density, cluster radial periods, etc., which cannot easily be broken down into a general galactocentric position scaling, only the internal dependences of these mechanisms are given:

- Dynamical friction, $\mu_{df} \propto M_c$ (e.g. section 1.4.2)
- Tidal shocking, $\mu_{sh} \propto M\rho_h^{-1} \propto r_h^3$ (e.g. section 1.4.3)
- Two-body relaxation driven evaporation, $\mu_{ev} \propto \rho_h^{1/2} \propto M^{1/2}r_h^{-3/2}$ (e.g. section 1.4.4)

One way to settle the contention of which one of these destruction mechanisms is responsible for carving out a GCMF from an initially power-law CIMF would be to observe intermediate age cluster systems. If the mass function is imprinted at an early age, it would be expected to resemble old, evolved functions; whereas if the mass function gradually changes as a result of slow decay, it would be expected to have a mass function somewhere between rising steadily towards low masses, and being distinctly peaked. This is currently an ongoing topic however, as identifying and making accurate observations of appropriately aged cluster populations is not trivial; knowledge of where to look is required, and a statistically significant number of objects to observe must be present (Chandar et al. 2006; Goudfrooij et al. 2004; Glatt et al. 2011; Goudfrooij 2012). Presently, data indicate that GCMF evolution is a gradual ongoing process rather than instantaneous, as displayed in Figure 1.10, showing the mass function of the GCS of NGC1316 in different ranges of half-mass density, ρ_h , and projected galactocentric radius, R_{gal} , as indicated in the top right of each panel (Goudfrooij 2012). The solid red line indicates the best fit power-law to the GCMF with $M > 10^5 M_\odot$ in that ρ_h or R_{gal} band. The dotted black curve displays the best fit power-law to the GCMF of the entire GCS for which $\beta_{MF} = 1.88 \pm 0.04$, and the dashed magenta line displays a model assuming evaporation dominated mass-loss for 3Gyr acting on an initially Schechter (1976) CIMF (these kinds of models will be described in much more detail in Chapter 2). Clearly the GCMFs made of higher ρ_h and lower R_{gal} clusters in the

bottom panels have been most strongly affected by mass-loss, as expected with the evaporation scaling, with a turn-off from a power-law regime becoming apparent at around $M \simeq 10^5 M_\odot$. Although it is expected that smaller R_{gc} would mean smaller r_M , which would lead to stronger tides (cf. equation 1.8) and thus higher ρ_h , the radially binned GCMFs in the right hand panels have more intermediary values for their power-law fits compared to the density binned GCMFs in the left hand panels. This is likely because of a lack of correlation in ρ_h vs R_{gc} , as was found in the Milky Way (McLaughlin & Fall 2008), M104 (Chandar, Fall & McLaughlin 2007), and in 22 Virgo galaxies in Chapter 2.

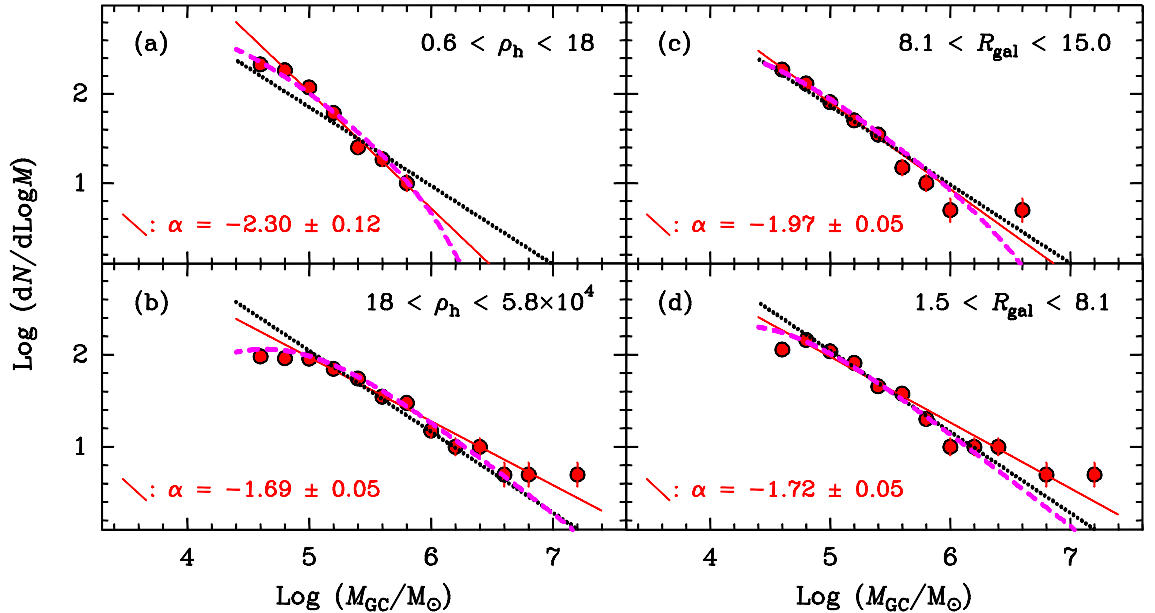


Figure 1.10: GCMF of the GCS of NGC 1316 divided up into bins of high and low half-mass density, ρ_h (in units of $M_\odot \text{pc}^{-3}$), and small and big projected galactocentric distance, R_{gc} (in units of kpc), taken from Goudfrooij (2012). See text for details.

The dependences of the three slow evolutionary processes can be used to define an area in $M - R_h$ parameter space known as a survival triangle. Outside of this triangle, clusters are expected to have either undergone or be undergoing destruction. By requiring that the sum of the reciprocals of these destruction rates (equations 1.19,

1.22, 1.25) at a given galactocentric radius with a given mass correspond to a destruction time-scale equal to a Hubble time, it is possible to define an area on the fundamental plane that should contain the majority of surviving GCs, as displayed in Figure 1.11.

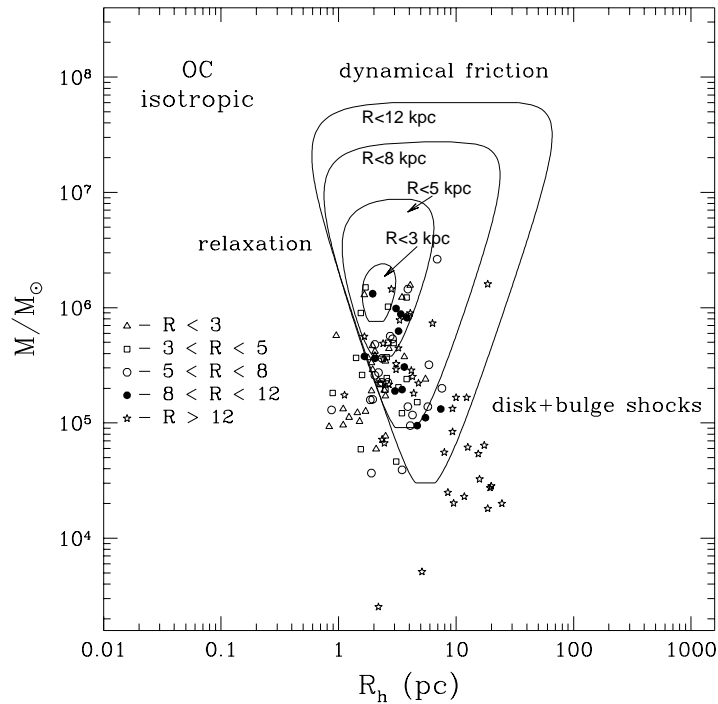


Figure 1.11: The survival triangle for Milky Way GCs (Gnedin & Ostriker 1997).

The fact that most of the data are crowded against the relaxation line, and are mainly distant from the dynamical friction or the tidal shocking sides suggests that it is evaporation that is most responsible for having driven the evolution of the observed extant GCs in the Milky Way, as otherwise, cluster populations in the process of being ‘culled’ by these mechanisms would also crowd against those boundaries (e.g., Fall & Rees 1977; Okazaki & Tosa 1995; Ostriker & Gnedin 1997; Gnedin & Ostriker 1997). As can be seen by the contours of triangles with different galactocentric distances, both dynamical friction and tidal shocking destruction time-scales show significant dependence on cluster galactocentric position. This is because dynamical friction depends

on the density of background stars surrounding the cluster, which lessens further away from the Galactic centre (cf. equation 1.19). Both disk and bulge shocking depend on galactocentric radius through either radial or azimuthal period, and either disk surface density or pericentric distance (cf. equation 1.22). Evaporation is mainly an internal effect however, depending primarily on cluster rate of relaxation; hence the contours on the relaxation side of the triangle are more compacted (cf. equation 1.25).

Assuming that evaporation is indeed the dominant mass-loss mechanism, since the rate of mass-loss due to evaporation is roughly constant (as borne out in many simulations, e.g., Lee & Ostriker 1987; Chernoff & Weinberg 1990; Lee, Fahlman & Richer 1991; Vesperini & Heggie 1997; Gnedin, Lee & Ostriker 1999; Baumgardt 2001; Giersz 2001; Baumgardt & Makino 2003; Trenti, Heggie & Hut 2007; Lamers, Baumgardt & Gieles 2010), the evaporation mass-loss rate may be written as:

$$\frac{d M_c}{dt} = -\frac{\xi M_c}{t_{rh}} \simeq \text{const} \quad (1.26)$$

which can be integrated to give:

$$\frac{1}{\xi} = \frac{t_{ev} - t}{t_{rh}} \quad (1.27)$$

where $t_{ev} - t$ is the expected remaining lifetime of a cluster, and t_{rh} its present relaxation time. Estimates of ξ then indicate how many more present relaxation times clusters are expected to last for. For tidally limited clusters, this is roughly 20-60, whereas for isolated clusters, this is 100-300 (e.g., Hénon 1961; Spitzer & Chevalier 1973; Spitzer & Shull 1975; Fall & Rees 1977; Caputo & Castellani 1984; Spitzer 1987; Aguilar, Hut & Ostriker 1988; Chernoff & Weinberg 1990; Gnedin & Ostriker 1997; Murali & Weinberg 1997; Gnedin, Lee & Ostriker 1999; Jordán et al. 2007; Chandar, Fall & McLaughlin 2007; McLaughlin & Fall 2008; Goudfrooij 2012).

1.5 Dynamical Evolution of the GCMF

Over the last few decades, much work has been done to calculate the combined effect of the destruction mechanisms discussed in section 1.4 on the evolution of the

GCMF. For example, Aguilar, Hut & Ostriker (1988) looked at cluster destruction rates due to evaporation, tidal shocking, and dynamical friction by fitting an assumed kinematic model for the GC system, and thereby generate initial orbital parameters for the clusters. They then integrated these orbits and calculated the destruction rates, and concluded that tidal shocking by the bulge is very efficient at destroying clusters on highly eccentric orbits, but that after these clusters are destroyed, tidal shocking is the least important mechanism with evaporation the dominant effect.

Okazaki & Tosa (1995) assumed an initial power-law cluster mass function, with $dN/dM \propto M^{-2}$, and evolved it for 13 Gyr including evaporation, disk shocking, and dynamical friction. They concluded that evaporation is the dominant mass-loss mechanism, and that dynamical friction only has an effect on the few innermost GCs and can be ignored. They also concluded that evaporation was capable of turning an initial power-law CIMF into a log-normal distribution as observed today. Furthermore, they looked at the evolution of the GCFP by assuming an initial mass-radius relation, but found that their resulting evolved GCFP largely depended their initial mass-radius relation.

Murali & Weinberg (1997) used Fokker-Planck models (a model described by a time-dependent distribution function) to study the disruption of star clusters by evaporation and gravitational shocks. They assumed a power-law CIMF, and modelled the evolution of clusters in a galaxy with a halo component alone, and with both a halo and disk component. They found that the GCVP tends to become more tangentially baised due to clusters on highly eccentric orbits being more easily destroyed, and that initially the spatial distribution of GCs traced the spatial distribution of the halo, but became more flattened at smaller radii due to stronger tidal forces there. However, they only simulated clusters in the mass range $M \in [10^5, 5 \times 10^6] M_{\odot}$, and therefore not much can be said about their results with regards to the evolved mass function resembling a log-normal GCMF after 13 Gyr of evolution.

Vesperini (1998) looked at GCMF evolution as being driven by evaporation, tidal shocking, dynamical friction, and stellar evolution using N-body simulations assuming a power-law initial cluster mass function or a log-normal initial mass function. He found

that a log-normal initial mass function can retain its shape during a Hubble time of evolution, whereas a power-law initial mass function will turn into a log-normal mass function after a Hubble time of evolution. However, his models predicted a much stronger GCMF dependence on galactocentric radius than is corroborated in observations. Additionally, he assumed that all clusters moved on circular orbits in a Keplerian potential, which is far from a realistic treatment.

Baumgardt (1998) assumed a $\beta_{MF} = -2$ power-law initial mass function, and that cluster mass-loss rate corresponds to the tides at orbital pericentre. With these assumptions, they investigated the effects of dynamical friction and evaporation on GCSs with different initial mass-radius relations, different tracer power-law slopes, and different values of constant velocity anisotropy. They found that the GCMF developed a peak and broadly resembled the present day GCMF of the Milky Way after a Hubble time for an initial tracer power-law profile with slope of $\alpha \simeq -4.5$ and with an initial velocity profile with anisotropy parameter $\beta \simeq 0.5$, though with some differences around the inner parts of the Galaxy. They also found little dependence of the evolution of the GCMF on the initial mass-radius relation of clusters. However, when Baumgardt (1998) applied the same model to M87, they found that it could not match the observed spatial distribution of the M87 GCS at all radii, either underestimating the number of clusters at outer radii ($\gtrsim 15\text{kpc}$) or overestimating the number of clusters at inner radii ($\lesssim 15\text{kpc}$), depending on how it was normalised.

Fall & Zhang (2001) developed models to investigate the effects of evaporation, tidal shocks and stellar mass-loss individually in a singular isothermal sphere with either constant or radially increasing initial GCS velocity anisotropy. They found that stellar evolution dominates mass-loss at early times, but does not change the overall shape of the GCMF because the same fraction of mass is removed in all clusters, regardless of initial cluster masses. They also found that tidal shocking made a significant impact on the high-mass end of the GCMF at late times, but that overall evaporation is the dominant driver of GCMF evolution. They concluded that evaporation dominated evolution may turn a power-law CIMF into a log-normal GCMF. Furthermore, they concluded that a radially increasing velocity anisotropy is required to match the

observed invariance of the GCMF with galactocentric radius. While more sophisticated than previous attempts to model the evolution of the GCMF, they still assumed that all clusters are tidally limited at pericentre, and only investigated the evolution in a singular isothermal sphere with a few different anisotropy profiles. Additionally, Vesperini et al. (2003) used a very similar method to model the GCMF evolution of M87, and found that the degree of anisotropy required to make the GCMF radially invariant was inconsistent with observations.

Fall & Zhang (2001) suggested that the difficulties in reproducing a radially invariant GCMF is because models used to simulate evolution all assume a spherical and static galactic potential, whereas in reality the galaxy will have undergone mergers and close encounters. Prieto & Gnedin (2008) used N-body simulations to model the evolution of the GCMF through evaporation, stellar evolution, and tidal shocking, including hierarchical merging. They found that mergers can scramble the orbits of GCs, such that GC mass ranges that may have been depleted within some range of galactocentric distance are repopulated. Thus they were able to produce an evolved GCS with a velocity distribution compatible with observations, and with a GCMF peak mass that matches observations and is the same at all galactocentric radii. However, they assumed in their model that all clusters had constant density, making inferring any conclusions about how a time-dependent potential might affect cluster destruction impossible.

However, Jordán et al. (2007) argued that rather than calculate mass-loss in GCs using calculated tidal density based on the strength of tides at some point along an orbit, it may be more natural to measure the density of GCs (either through fits of King (1966) models, or measurements of half-light radii) directly, and use these to calculate mass-loss. Thus if clusters did indeed have their densities set by tides at orbital pericentre, but later had their orbits scrambled such that their new orbital pericentres no longer correspond to their densities, it would not matter as the mass-loss history can still be calculated directly from the density. Furthermore, orbiting bodies move most slowly at apocentre, and hence this is where they are most likely to be observed. Therefore, if rather than calculate the orbit, the present galactocentric

radius is used to calculate the tidal density, yet more error is stacked up. In galaxies other than the Milky Way, projection effects make the problem even worse. By using internal cluster densities, all of these problems are avoided. By assuming constant mass-loss rate due to evaporation and a Schechter (1976) CIMF, Jordán et al. (2007) built very simple models for evaporation dominated mass-loss, where the average mass lost for a group of clusters, Δ , is a fitting parameter, C , times the duration of the evolution, t , times the square root of the mean cluster densities in that group, $\bar{\rho}_h$, i.e. $\Delta = C\bar{\rho}_h^{-1/2}t$. They were then able to combine this average mass-loss with the CIMF to make an evolved GCMF as a function of Δ . By fitting these models to data, they were then able to obtain a best fit value for the parameter C , and found it to be in good agreement with theoretical results and simulations (see section 1.2.2). McLaughlin & Fall (2008) improved on this procedure by using individual cluster densities, rather than the mean densities of a group of GCs.

Given the mounting evidence indicating that GCMF evolution is indeed dominated by evaporation, in the next Chapter the types of simple single parameter models as derived by McLaughlin & Fall (2008) are used to obtain an estimate for C using data from 22 early-type Virgo galaxies. Chapter 3 then describes the physics, numerics, and procedures for setting up a simulated GCS with specifiable initial spatial, kinematic, and mass distributions in a specifiable host galaxy model, and ends by describing a code utilising these physics, numerics and procedures. Chapter 4 displays the output of this code for a wide variety of GCS properties in a range of host galaxy potentials. Chapter 5 then proceeds to describe the physics, numerics, and procedure for then taking these simulated initial GCSs and evolving them through time with a choice of any combination of evaporation, tidal shocking, dynamical friction, and stellar evolution, each represented by a choice of several prescriptions taken from the literature, and using results from Chapter 2. Chapter 6 begins with the description of a second code capable of taking the initial simulated GCSs output by the first code, and applying the evolutionary methods and prescriptions described in the previous Chapter to produce evolved simulated GCSs. Demonstrations of the application of this second code with a selection of evolutionary methods and prescriptions to a variety of initial GCSs are

then presented. This is then followed by the discussion and conclusions in Chapter 7.

2 GCMFs in Early-Type Virgo Galaxies

2.1 Introduction

With current technology it is possible to measure internal GC properties such as velocity dispersions, metallicities, etc., throughout the Local Group (e.g., Barmby, Holland & Huchra 2002; McLaughlin & van der Marel 2005; Barmby et al. 2007). These properties may be used to piece together an understanding of GC evolution, thus giving clues to cluster formation, and by extension, star formation in the early universe. In order to build as comprehensive a picture of GC evolution as possible, as many data as possible are required. However, acquiring these sorts of GC data outside of the Local Group is extremely difficult if not impossible. Until technology advances sufficiently to directly measure internal GC properties in galaxies further afield, it is still possible to measure some basic properties, such as GC sizes (e.g., Jordán et al. 2005) and luminosities (e.g., Jordán et al. 2006; Jordán et al. 2007) across an entire GCS. By looking at the number of globular clusters per unit luminosity, (the globular cluster luminosity function, GCLF) or analogously, the number of globular clusters per unit mass (the globular cluster mass function, GCMF — see section 2.2.2) can be delineated.

Previous attempts to model the evolution of GCMFs have revolved around inferring the internal density of clusters via their orbital pericentres. These attempts have then resulted in predicting a radial dependence of the GCMF that is contrary to observations (see section 1.5). Rather than attempt to build models predicting a radially invariant GCMF, Jordán et al. (2007) instead developed models depending on internal cluster densities. These models assumed an initial cluster mass function (CIMF) in the form of power-laws, resembling the mass functions of young massive clusters as observed in mergers (e.g., Whitmore & Schweizer 1995; Schweizer et al. 1996; Zhang & Fall 1999; Fall & Zhang 2001; Fall, Chandar & Whitmore 2005; Bastian et al. 2006), and roughly constant evaporation dominated rate of mass-loss. By deriving an expression relating current and initial GC mass depending on evaporation rate, Jordán et al.

(2007) were able to evolve these CIMFs over 13 Gyr to create mass functions resembling GCMFs as observed in the Virgo cluster. In this Chapter, the same data published in Jordán et al. (2009) are used in similar but more advanced models to obtain an estimate for a parameter directly related to ξ , the fraction of mass lost per relaxation time (cf. equation 1.27), giving $\xi \simeq 0.078 \pm 0.016$, indicating that GCs in the Virgo system are expected to survive for on average about another 13 relaxation times before being destroyed by the tidal fields of their host galaxies, in good agreement with other studies (e.g., Hénon 1961; Spitzer & Chevalier 1973; Spitzer & Shull 1975; Fall & Rees 1977; Caputo & Castellani 1984; Spitzer 1987; Aguilar, Hut & Ostriker 1988; Chernoff & Weinberg 1990; Gnedin & Ostriker 1997; Murali & Weinberg 1997; Gnedin, Lee & Ostriker 1999; Jordán et al. 2007; Chandar, Fall & McLaughlin 2007; McLaughlin & Fall 2008). Furthermore, the GCMF dependence on average internal cluster density, whilst simultaneously lacking any strong dependence on galactocentric position is reaffirmed, in agreement with studies conducted in the Milky Way (McLaughlin & Fall 2008), M104 (Chandar, Fall & McLaughlin 2007), and NGC1316 (Goudfrooij 2012). The overall aim of this Chapter is to test the hypothesis that evaporation may turn a power-law CIMF into a GCMF with a distinct peak that is constant with radial position as observed in the Milky Way and in other extant GCSs.

Theory predicts the relation $\mu_{ev} \propto \rho_h^{1/2}$, where μ_{ev} is mass-loss rate due to evaporation, and $\rho_h = 3M/(8\pi r_h^3)$ is half-mass density. However this simple relation is complicated by additional considerations such as the ‘apocentre criterion’, where escaping stars must pass beyond the tidal radius before being lost from the cluster, the oscillating tidal radius of a GC on an elliptical orbit, and even the internal GC structure (see section 1.4.4). Accounting for any one of these additional considerations is a difficult and complicated problem, let alone attempting to derive a general expression for μ_{ev} . However, N-body simulations have shown that mass-loss due to evaporation is roughly linear with time for the majority of a cluster’s lifetime for a wide variety of cluster structures (Baumgardt & Makino 2003). Thus it is possible to take the simple relation $\mu_{ev} \propto \rho_h^{1/2}$, assume that evaporation is the dominant cause of mass-loss, and fit for the proportionality constant. In doing so, the additional uncertainties in evapora-

tion rate due to the apocentre criterion, oscillating tidal radii, and cluster structure are absorbed into this proportionality constant. If these effects significantly modified the mass-loss rate from a simple dependence on half-mass density, we would expect that assuming $\mu_{ev} \propto \rho_h^{1/2}$ and fitting for the proportionality constant would provide a poor fit to observed data. However, McLaughlin & Fall (2008) showed that equally good fits are obtained in the Milky Way when mass-loss rates are taken to scale with tidal densities as implied by King (1966) model fits rather than half-mass densities, and in fact found that their constant of proportionality changed by a constant factor of 210, the median of GC ρ_h/ρ_t in the Harris (1996) catalogue. Furthermore, McLaughlin & Fall (2008) also showed that GCMF models of very similar shape can be fitted, with equally good results, if evaporation rates are taken to scale with tidal surface densities, $\Sigma_t \propto M/\pi\rho_t^{1/2}$ (a scaling closer to what is expected for evaporation modified by the apocentre criterion). Therefore, since fitting King (1966) models to GCs in Virgo provide tidal radii so uncertain as to be essentially meaningless, it is assumed that the findings of McLaughlin & Fall (2008) in the Milky Way also apply in each of the 22 Virgo galaxies.

Before proceeding to fit half-mass density-dependent models to observed GCMFs, a further point to consider is that mass segregated clusters have constituent star masses correlated with distance from the cluster centre (see section 1.4.5). Since the dependence of star luminosity on stellar mass is highly non-linear, the luminosity and mass profile of mass segregated clusters are consequently different. As many GCs are thought to be mass segregated (e.g., King, Sosin & Cool 1995; Ferraro et al. 1997; Andreuzzi et al. 2000; Howell, Guhathakurta & Tan 2000; Koch et al. 2004), this means that generally half-mass and half-light radii will be different, $\rho_{h,light} \neq \rho_{h,mass}$. However, this is not a big problem for old clusters, as low-mass stars are preferentially shed during evaporation (e.g., Lamers, Baumgardt & Gieles 2013) and high-mass stars expire through stellar evolution, resulting in older clusters having a narrower range of stellar masses (and thus luminosities) present. Hence, old clusters tend towards single mass clusters, and the half-mass and half-light radii converge. Thus using half-light densities in place of half-mass densities in the GCMF models should only have a minor affect

on the models.

A further important consideration for this investigation is whether Virgo GCs are actually tidally limited at all. Due to the difficulties in calculating tidal radii of GCs outside of the MW (i.e. either contending with projection when trying to infer a tidal density from observed galactocentric positions, or in fitting King (1966) models to clusters whose light profiles are barely visible outside of the effective radius), it is simply assumed that all GCs in the 22 galaxies are tidally limited and have been for the majority of their lifetimes. Thus, these Virgo GCs will have been undergoing an approximately constant rate of mass-loss for the majority of their lives due to evaporation. However, other studies bring into question whether GCs can accurately be assumed to be tidally limited at all. For example, Baumgardt et al. (2010) identified two populations outside of 8 kpc in the Milky Way GCS based on a plot of r_h/r_J against r_{gc} (their Figure 2), revealing two distinct populations; a ‘compact’ population ($r_h/r_J \lesssim 0.05$) and a ‘tidally limited’ population ($0.1 \lesssim r_h/r_J \lesssim 0.3$). Since $r_h/r_t \lesssim 0.1$ is undefined for King (1966) models (cf. Figure 1.7), they concluded that these ‘compact’ clusters must be under-filling their Roche lobes. However, Baumgardt et al. (2010) used present GC galactocentric radii to calculate r_J when computing r_h/r_J . Given that orbiting bodies move most slowly near apocentre, this is where they are statistically most likely to be observed. Since the tidal density of a GC is actually expected to be set somewhere near pericentre, Baumgardt et al. (2010) used very inflated r_J values. The ratio of pericentre, r_p , to apocentre, r_a , is related to orbital ellipticity via $e = (1 - \frac{r_p}{r_a}) / (1 + \frac{r_p}{r_a})$. In a singular isothermal sphere the average ellipticity is around $e \simeq 0.55$, largely independent of velocity anisotropy (van den Bosch et al. 1999), giving the ratio of apocentre to pericentre to be about $r_p/r_a \simeq 0.29$. Using the same equation for r_J as Baumgardt et al. (2010),

$$r_J = \left(\frac{GM_c}{2V_G^2} \right)^{1/3} r_{gc}^{2/3} \quad (2.1)$$

this amounts to a difference of a factor of ~ 0.44 in r_J . This would go a long way towards, and in many cases resolve the problem of r_h/r_J not falling within the allowed

range of r_h/r_t for King (1966) models for these ‘compact’ clusters.

Additionally, Gieles et al. (2011) analytically derived models describing the evolution of cluster density, half-light radius, relaxation time, and crossing time as functions of cluster evolution (the ratio of current age to total lifetime, t/τ_{ev0} in their notation). By taking the time derivative of their expression for relaxation time and finding the value of t/τ_{ev0} corresponding to the stationary point, Gieles et al. (2011) then defined clusters with ratios of age to total lifetime below this value as expansion dominated, and those above as evaporation dominated (i.e. separating those that are tidally limited, and those that are still in the process of expanding to fill their Roche lobes), and concluded that 93 of the 141 Milky Way clusters in their sample are not tidally limited. However, this is an ad-hoc result as their models have the built-in assumption that clusters are born severely under-filling their tidal radii (i.e. essentially in isolation). In the derivation of their models this assumption is manifested where they assume that the initial crossing time of newborn clusters is zero, equivalent to an infinite initial density. Realistically, some clusters could form already filling a significant fraction of their Roche lobe (e.g., Elson, Fall & Freeman 1987), thus decreasing the fraction of their total lifetimes taken to reach evaporation dominated evolution (i.e. the position of the stationary point in relaxation time would be at lower t/τ_{ev0}), and thus fewer clusters would fall under the isolated regime. Indeed, in some cases clusters may even be born tidally limited. In any event, Gieles et al. (2011) assumed a constant rate of mass-loss throughout cluster evolution, which is the only reason the assumption that the 22 Virgo galaxy clusters are tidally limited is made in the first place. Consequently there is little actual evidence that Milky Way GCs are anything but tidally limited, and we proceed to assume the same of GCs in Virgo.

Having addressed the technical issues in the assumptions and methodology, the next section details the data, followed then by the derivation of the density-dependent GCMF models. Following the models are the results and conclusions about the investigation.

2.2 Data

This investigation utilises data collected as part of the HST ACS Virgo Cluster Survey (Côté et al. 2004). The survey imaged 100 early-type Virgo galaxies in deep F475W (denoted W throughout) and F850LP (denoted LP throughout) for a total of 750 and 1210 seconds respectively. Jordán et al. (2009) presented these data, giving projected galactocentric positions (R_{gc}), foreground $E(B - V)$ ($A_W = 3.634E(B - V)$, $A_{LP} = 1.485E(B - V)$; Jordán et al. 2004), King (1966) projected half-light radii ($R_{h,W}$, $R_{h,LP}$), magnitudes (m_W , m_{LP} , $M_{\odot,W} = 5.13$ and $M_{\odot,LP} = 4.56$; Gonzaga 2011), and background flux ($I_{b,W}$, $I_{b,LP}$) for the GCs of each galaxy imaged (Jordán et al. 2004). Note that by numerically projecting spherical GC models (e.g., King 1966 models), it is possible to calculate the unique projected half-light radius, R_h , corresponding to the unprojected half-light radii, r_h . Thus by assuming a spherically symmetric GC, it is possible to also calculate r_h given R_h . For King (1966) models, the relation $r_h = 4R_h/3$ holds for a large range of the concentration parameter c , and so this is how projected half-light radii are deprojected throughout this Chapter.

Jordán et al. (2009) then developed a ‘clustering method’, which was utilised to assign a probability that each source is a bona fide GC: those with a probability less than 0.5 were removed in an attempt to remove contaminants such as foreground stars or background galaxies. Furthermore, Jordán et al. (2009) culled the data in terms of an upper limit on apparent magnitude in a further attempt to remove contaminants. An additional cut to the data was implemented in this investigation by requiring that all data satisfy $|R_{h,W} - R_{h,LP}| < (R_{h,W} + R_{h,LP})/2$, as the W and LP radii of some data differed by as much as a factor of 3, much more than could be accounted for with typical random uncertainties (± 0.0003 arcseconds ≈ 0.25 pc; Jordán et al. 2005). Throughout this investigation, LP -band observations were preferentially used, as LP -band mass-to-light ratios (Υ_{LP}) are much less sensitive to metallicity than W -band, reducing the scope for additional error in cluster masses (e.g., Jordán et al. 2007, see Figure 2.1). The reason that LP mass-to-light ratios are insensitive to metallicity is that the LP bandpass is at longer wavelengths than W (850nm vs 475nm peak response

throughput), and the spectra of different metallicity simple stellar populations tend to differ little at longer wavelengths (e.g., Schulz et al. 2002).

In order to ensure good number statistics in the observed GCMFs, GC data were only used from galaxies with observed cluster populations of at least 130 in the Jordán et al. (2009) catalogue. This resulted in a selection of 26 galaxies, 4 of which are compact ellipticals associated with either M49 (VCC 1226) or M87 (VCC 1316) (Jordán et al. 2007; Jordán et al. 2005), namely VCC 1327, VCC 1297, VCC 1192, and VCC 1199. These 4 were excluded, as the majority of GCs in their vicinity likely actually belong to M49 or M87. Images were then available for each of the remaining 22 galaxies' GCs in both W and LP bandpasses. Absolute magnitudes were calculated for the GCs with distance moduli measured by Mei et al. (2007) using surface brightness fluctuations, although some could not be calculated due to dust excess. Where dust excess was a problem, a distance modulus of $(m - M) = 31.1$ was assumed (Mei et al. 2005) (equivalent to 16.6 Mpc).

2.2.1 Incompleteness

At the distance of Virgo, photometric data incompleteness can become an issue. This is caused by such factors as clusters being missed due to being too dim/distended, or due to being obscured by their host galaxies. Jordán et al. (2009) produced completeness tables, giving the probability that a cluster with a given magnitude, m , half-light radius, R_h , and against a given background intensity, I_b , is observed. These completeness tables were produced by inserting artificial clusters with known m and R_h into an image with known I_b , and recording the fraction of artificial clusters recovered by their data pipeline. This fraction is then equivalent to the probability that such a cluster against such a background is observed.

Without correcting for incompleteness, any model of GCMF evolution may predict the presence of clusters in areas where there is data missing due to incompleteness, resulting in a bad fit. Thus in both W and LP , the completeness tables produced by Jordán et al. (2009) were interpolated over and convolved with the GCMF evolution

models (to be described in section 2.3), such that the GCMF model predicted accordingly fewer GCs where they are less likely to be observed.

Additionally, in the interests of limiting the model GCMF predictions in the same way that the data was limited, it was necessary to account for the data cuts imposed by Jordán et al. (2009). Their criteria are as follows:

- $0.5 \text{ mag} < W - LP < 1.9 \text{ mag}$
- $0.75 \text{ pc} < r_h < 10 \text{ pc}$
- $LP < 25.15 \text{ mag}$
- $W < 26.35 \text{ mag}$

This was accomplished by simply setting the aforementioned completeness probability to be convolved with the GCMF models for any clusters that did not satisfy these criteria to zero, regardless of what the completeness probability was actually tabulated as.

2.2.2 Mass-to-Light Ratios

Mass-to-Light ratios for W and LP (Υ_W and Υ_{LP}) were calculated using population synthesis models for every cluster and used to convert from luminosity to mass, thus also allowing conversion from a GCLF to a GCMF. The population synthesis models used were Bruzual & Charlot (2003) with appropriate HST F475W and F850LP filter response curves (ACS Handbook, Gonzaga 2011) added to the filter database. The assumed IMF was the Chabrier (2003) disk IMF, with an assumed age of 13 Gyr for all GCs, leading to curves of Υ_W and Υ_{LP} against $W - LP$ colour (e.g. see Figure 2.1). Υ_W and Υ_{LP} were then interpolated for every individual cluster using observed W and LP magnitudes.

Curves of Υ_W and Υ_{LP} were also computed with the Maraston (2005) population synthesis models to facilitate comparisons, and were found to be in good agreement

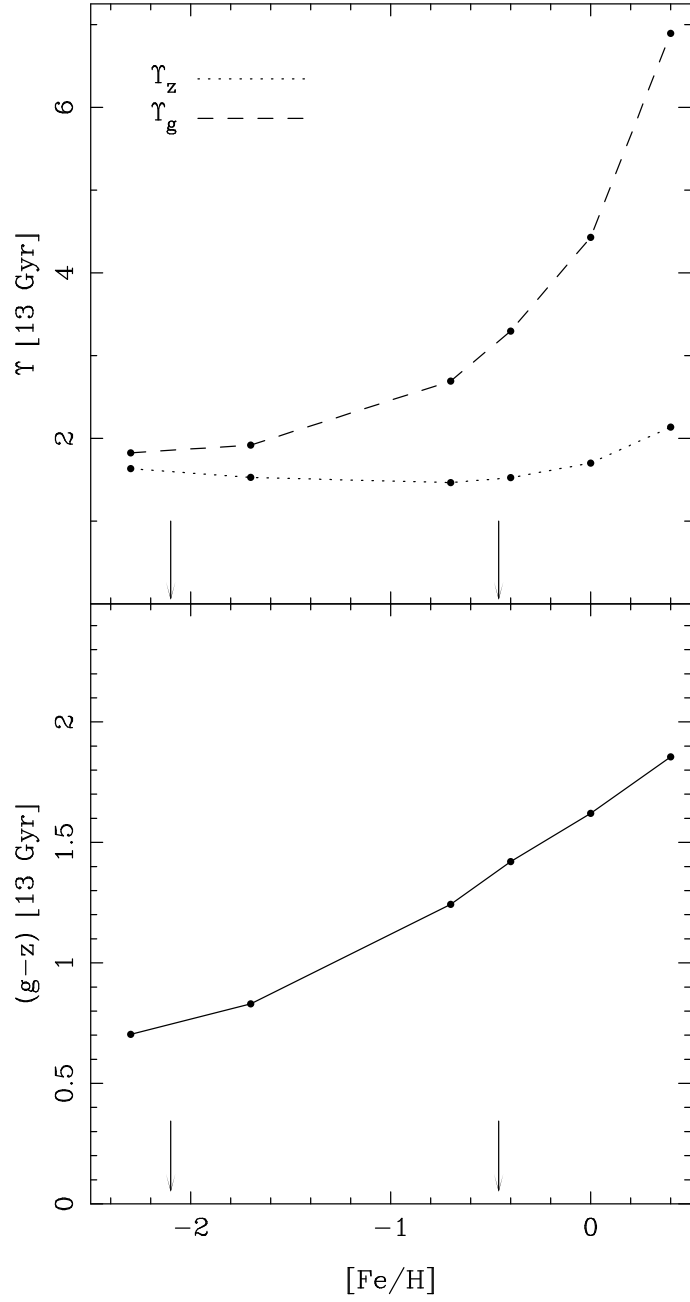


Figure 2.1: Top panel: mass-to-light ratios in solar units in the F475W (denoted g in this plot) and F850LP (denoted z in this plot) bandpasses against metallicity. Bottom panel: F475W - F850LP colour against metallicity, all as predicted by the PEGASE population synthesis model for a 13 Gyr old simple stellar population (Fioc & Rocca-Volmerange 1997). The arrows indicate the maximum and minimum average GC $[\text{Fe}/\text{H}]$ in the ACSVCS galaxies. Plot taken from Jordán et al. 2007.

with the Bruzual & Charlot (2003) models. Υ_W and Υ_{LP} were then multiplied by a corrective factor of 0.8 to correct for the preferential loss of low-mass stars during the process of relaxation driven evaporation raising the average stellar mass, and consequently the average stellar luminosity even more (e.g., McLaughlin & van der Marel 2005; Baumgardt & Makino 2003; McLaughlin & van der Marel 2005; Barmby et al. 2007; Lamers, Baumgardt & Gieles 2013). The masses calculated using W and LP magnitudes combined with these mass-to-light ratios were in good agreement, with an average deviation of 10%. Kruijssen & Lamers (2008) developed models in V -band to describe the evolution of average stellar mass in a cluster as a function of age (and hence, mass-to-light ratio) by considering the opposing effects of stellar evolution depleting the high-mass end of the stellar mass function, and the preferential depletion of low-mass stars by evaporation. By also treating for mass-dependent escape rates during evaporation, they theorised that this corrective factor to the mass-to-light ratio should itself be a function of cluster mass, which would cause the shapes of the GCMF and GCLF to differ, rather than simply being rescaled versions of one-another. In order for the preferential loss of low-mass stars to be significant, a cluster must first be mass segregated. Since the time-scale for mass segregation is the relaxation time, it is therefore expected that at a given density, lower mass clusters with shorter relaxation times will be more adversely affected by mass-dependent mass-to-light ratios. Kruijssen & Lamers (2008) predicted that this effect could amount to as much as 0.6 dex for clusters with luminosities $L_V \simeq 10^4 L_\odot$, but is negligible for clusters with luminosities $L_V \gtrsim 10^{5.5} L_\odot$ (e.g. their Figure 7). Incorporating these models would require the detailed mass-loss history of each individual cluster, which would erase the benefit of the simple density-dependent GCMF models derived next. Additionally, since this consideration will most strongly affect the low-mass end of the GCMF which already has problems with incompleteness anyway, it is therefore neglected.

2.3 Density-Dependent GCMF Models

These models begin by assuming a Schechter (1976) CIMF, giving the probability of a cluster having an initial mass between M_0 and $M_0 + dM_0$:

$$\frac{dN}{d \log M_0} \propto M_0^{1-\beta_{MF}} \exp \left[-\frac{M_0}{M_{sch}} \right] \quad (2.2)$$

where an exponent of $\beta_{MF} \approx 2$ gives good fits to young cluster mass functions (e.g., Zhang & Fall 1999; Chandar et al. 2010; Whitmore et al. 2002; Gieles et al. 2006; Bastian et al. 2007; Jordán et al. 2007), and $M_{sch} \approx 10^6 M_\odot$ describes at what mass the exponential cut-off in the GCMF occurs, also constrained by observations of real cluster mass functions (e.g., Jordán et al. 2007; Harris et al. 2009).

Assuming that evaporation is the dominant mass-loss mechanism over a cluster lifetime (see section 1.4 for an overview of GCMF evolution mechanisms), and that evaporation rate is roughly constant over that time, then since mass-loss due to evaporation is a constant fraction per relaxation time, $\mu_{ev} = -\xi M_c / t_{rh} \propto -\rho_h^{1/2} = -C \rho_h^{1/2}$ where C is the proportionality constant (see section 1.4.4). Note that this implies C has units of $M_\odot^{1/2} \text{pc}^{3/2} \text{Gyr}^{-1}$. This expression for μ_{ev} may be integrated from $t = 0$ to the present time to give the following relation between initial and present mass:

$$M_0 = M_c + C \rho_h^{1/2} t \quad (2.3)$$

Alternatively, the integration may be taken from the present time to the time when the cluster has completely dissolved, t_{ev} , leading to the same expression as in equation (1.27), relating the fraction of mass lost per relaxation time, ξ , to the number of present relaxation times a cluster has left before destruction, $(t_{ev} - t)/t_{rh}$. Thus, physically the parameter C contains information about the expected lifetimes of clusters, and is related to ξ through:

$$\xi = \frac{0.138C}{\bar{m} \ln(\lambda N)} \left(\frac{3}{8\pi G} \right)^{1/2} \quad (2.4)$$

where \bar{m} is the mean stellar mass in a cluster, and $\ln(\lambda N)$ is the Coulomb logarithm.

Simple application of the chain rule and conservation of probability allows the CIMF to be related to the evolved cluster mass function (Fall & Zhang 2001):

$$\frac{dN}{d \log M_c} = \frac{dN}{d \log M_0} \frac{\partial \log M_0}{\partial \log M_c} = \frac{M_c}{M_0} \frac{dN}{d \log M_0} \frac{\partial M_0}{\partial M_c} = \frac{M_c}{M_0} \frac{dN}{d \log M_0} \quad (2.5)$$

Then substituting equations (2.2) and (2.3), the resulting equation gives the probability that a cluster with density ρ_h has a mass between M and $M + dM$ at time t (Jordán et al. 2007):

$$\frac{dN}{d \log M_c} = \frac{A M_c}{(M_c + C \rho_h^{1/2} t)^{\beta_{MF}}} \exp \left[-\frac{M_c + C \rho_h^{1/2} t}{M_{sch}} \right] \quad (2.6)$$

Where A is a normalisation constant. As this is a probability distribution, the integral of this function from the lowest possible cluster mass ($M_l = 10^2 M_\odot$) to the highest ($M_u = 10^8 M_\odot$) is unity, and thus the normalisation constant is given by:

$$A = \ln(10) \left(\int_{M_l}^{M_u} (M_c + C \rho_h^{1/2} t)^{-\beta_{MF}} \exp \left[-\frac{M_c + C \rho_h^{1/2} t}{M_{sch}} \right] dM_c \right)^{-1} \quad (2.7)$$

The peak of the cluster probability distribution is analytically calculable and is given by:

$$M_{TO} = \frac{M_{sch}}{2} \left[- \left(\frac{C \rho_h^{1/2} t}{M_{sch}} + \beta_{MF} - 1 \right) + \sqrt{\left(\frac{C \rho_h^{1/2} t}{M_{sch}} + \beta_{MF} - 1 \right)^2 + \frac{4 C \rho_h^{1/2} t}{M_{sch}}} \right] \quad (2.8)$$

In order to account for data incompleteness, the probability distribution must also incorporate the chance that a cluster of magnitude m , half-light radius R_h and against a background of intensity I_b is not observed. This is done simply by multiplying the probability distribution and the completeness fraction (see section 2.2.1), such that the product gives the probability that a cluster with mass M_c , density ρ_h , magnitude m , half-light radius R_h , and against a background of intensity I_b exists at time t , and is observed. Using distance moduli (Mei et al. 2005; Mei et al. 2007) and mass-to-light

ratios (see section 2.2.2), the magnitude may be turned to a mass, and with the mass already specified, the effective radius may be turned to a density. Thus the observable probability distribution actually only depends on M_c , ρ_h , and I_b , and is given by:

$$\frac{dN}{d \log M_c} = \frac{A M_c}{(M_c + C \rho_h^{1/2} t)^{\beta_{MF}}} \exp \left[-\frac{M_c + C \rho_h^{1/2} t}{M_{sch}} \right] \times f(M_c, \rho_h, I_b) \quad (2.9)$$

with A now given by:

$$A = \ln(10) \left(\int_{M_l}^{M_u} (M_c + C \rho_h^{1/2} t)^{-\beta_{MF}} \exp \left[-\frac{M_c + C \rho_h^{1/2} t}{M_{sch}} \right] f(M_c, \rho_h, I_b) dM_c \right)^{-1} \quad (2.10)$$

The total mass function of an entire GCS is then just the sum of each constituent cluster's probability distribution (McLaughlin & Fall 2008), i.e.

$$\frac{dN}{d \log M_c} = \sum_{i=1}^{\mathcal{N}} \frac{A_i M_{c,i}}{(M_{c,i} + C \rho_{h,i}^{1/2} t_i)^{\beta_{MF}}} \exp \left[-\frac{M_{c,i} + C \rho_{h,i}^{1/2} t_i}{M_{sch}} \right] \times f(M_{c,i}, \rho_{h,i}, I_{b,i}) \quad (2.11)$$

where \mathcal{N} is the total number of GCs in the GCS, and $M_{c,i}$, $\rho_{h,i}$, $I_{b,i}$ and t_i are the mass, density, background intensity, and age of the i 'th cluster. Both β_{MF} and M_{sch} are assumed to be parameters of the host galaxy and consequently the same for all clusters, and all clusters are assumed to be 13 Gyr old.

Assuming that $C \rho_h^{1/2} t \ll M_{sch}$ (i.e. that the mass lost from a cluster with density ρ_h at time t is much less than the Schechter (1976) exponential drop-off mass), it can be shown that $M_{TO} \simeq C \rho_h^{1/2} t / (\beta_{MF} - 1)$ if $\beta_{MF} > 1$. Hence there is a degeneracy between C and β_{MF} , making fitting for both as free parameters redundant. Furthermore, Jordán et al. (2006) and Jordán et al. (2007) analysed the GCLFs of the ACSVCS galaxies, fitting them with both Gaussian curves and ‘‘Evolved Schechter models’’ similar to those described above. They found that the dispersion of Gaussian models, and M_{sch} in the Schechter models, falls off with decreasing host galaxy luminosity, resulting in a range of M_{sch} values of $(2 - 3) \times 10^6 M_\odot$ for the brightest galaxies to $(3 - 4) \times 10^5 M_\odot$ for the faintest. Despite this systematic dependence, the origins of M_{sch} are

somewhat mysterious; so far there has been little physical explanation or reasoning for its existence. Indeed $M_{sch} \sim 10^6 M_\odot$ is entirely consistent with power-law CIMFs, but is rarely necessary for a good fit. The sole purpose for the existence of M_{sch} in these models is to allow the evolved GCMF curves to match the high-mass end of observed GCMFs. Consequently rather than for each galaxy fit for an additional parameter that is poorly understood, M_{sch} is fixed at the same value for all 22 ACSVCS galaxies. However, different values of M_{sch} are trialled, with the results reported in section 2.4.

Thus these are models of only a single variable parameter, C . The procedure for obtaining C is first to select a galaxy, and divide its GCS into bins of density. In order to keep a significant number of GCs in each division of density, only 3 bins were created. A wide range of values for C is then looped over, and equation (2.11) is then simultaneously applied to each density bin at each iteration of C , using the half-mass densities of the constituent GCs. At every iteration of C , values of χ^2 are calculated to compare the model GCMFs to the observed data in each bin, and the sum of χ^2 from each bin is recorded, where χ^2 is given by:

$$\begin{aligned}\chi^2 &= \sum_{i=1}^n \frac{(P_i - O_i)^2}{\sigma_i^2} \\ \chi_\nu^2 &= \frac{\chi^2}{n - 1}\end{aligned}\tag{2.12}$$

where n is the number of degrees of freedom (in this case the number of mass bins the GCMF was divided into), P_i is the predicted number of GCs at the mass of the i 'th bin, O_i is the observed number of GCs in the i 'th bin, σ_i is the error in the observed number in the i 'th bin, and χ_ν^2 is the reduced chi-squared.

The next value of C is then iterated, and the process is repeated until the loop completes. The optimal value for C is then selected through χ^2 minimisation, i.e., the value of C corresponding to the iteration giving the smallest sum of χ^2 values. Thus each galaxy has a value of C common to each density bin, and any differences between the bins is due to the different half-light densities of the constituent GCs in each bin. The fits giving the smallest sum of χ^2 are displayed in Figure 2.2.

Once C is obtained, the GCS is then subdivided into bins of galactocentric radius instead of half-light density, and equation (2.11) is applied using the recently acquired C and the densities of the clusters in each bin, as displayed in Figure 2.3.

Table 2.1 contains the details of each panel of Figures 2.2 and 2.3. The table is sorted by descending galaxy B -band luminosity (column 2), and lists the effective radius of each galaxy (column 3) and the number of GCs in each galaxy (column 4). Next is the median half-light density of each division of the density-sorted GC data (column 5). By fitting models according to the procedure outlined in section 2.3, the best-fit model turnover-masses (column 6), χ^2 (column 7), and fitting parameter C (column 8) are gained. The same value of C is then used with data sorted by R_{gc} (the median values of which are given in column 9), to produce models with turnover-masses listed in column 10, and χ^2 values listed in column 11.

2.4 Results

Figure 2.2 shows the models produced using equation (2.11) fitted to the GCS in each of the 22 galaxies, in order of descending galaxy luminosity. The only parameter varied in the fitting was the constant of proportionality, C , from equation (2.3). The three panels for each galaxy show the different cluster density bins, with the least dense clusters in the top panel, increasing downwards. The red dashed curve represents the model prediction for the intrinsic GCMF (i.e., if all GCs were visible), given by equation (2.11) with $f(M_{c,i}, \rho_{h,i}, I_{b,i}) = 1$ for all $M_{c,i}$, $\rho_{h,i}$, and $I_{b,i}$. The solid blue curve is generated by equation (2.11), displaying the predicted intrinsic GCMF convolved with incompleteness data, and is thus the prediction for the observable GCMF of that galaxy. Black points are GC data with Poisson error bars.

The reason that the red and blue curves deviate at the high-mass end of the top panel for every galaxy is due to the upper limit in r_h of 10pc imposed on the GCMF evolution models to match the cuts to the data made by Jordán et al. (2009). As the top panel contains the least dense clusters, any massive clusters must have

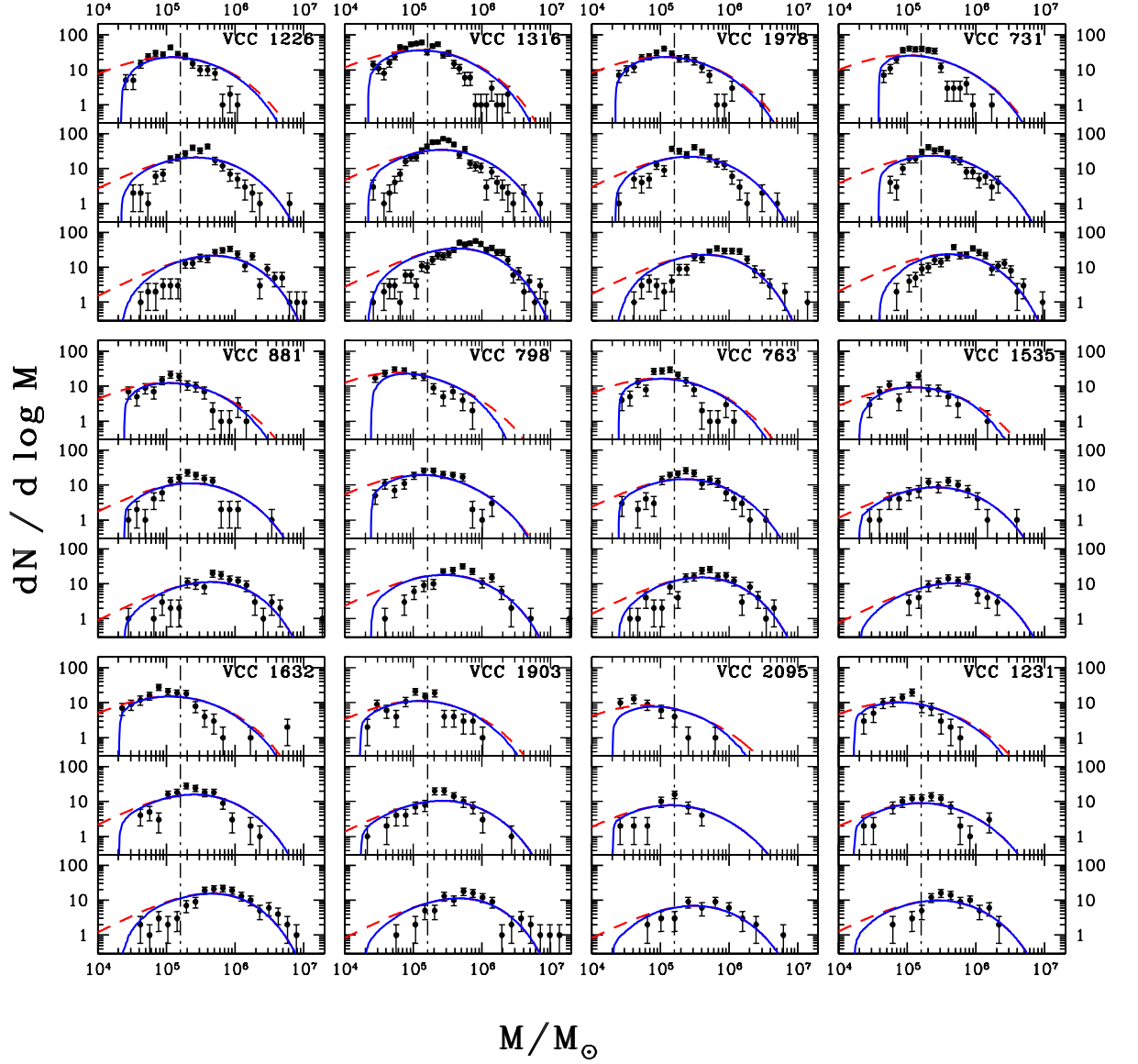


Figure 2.2: Observed mass function of the 22 ACSVCS galaxies selected in this sample, with evolved Schechter (1976) function (red dashed) and completeness degraded Schechter (1976) function (solid blue) applied to density sorted data. Galaxies are ordered by descending B -band luminosity. Details of each plot are given in Table 2.1.

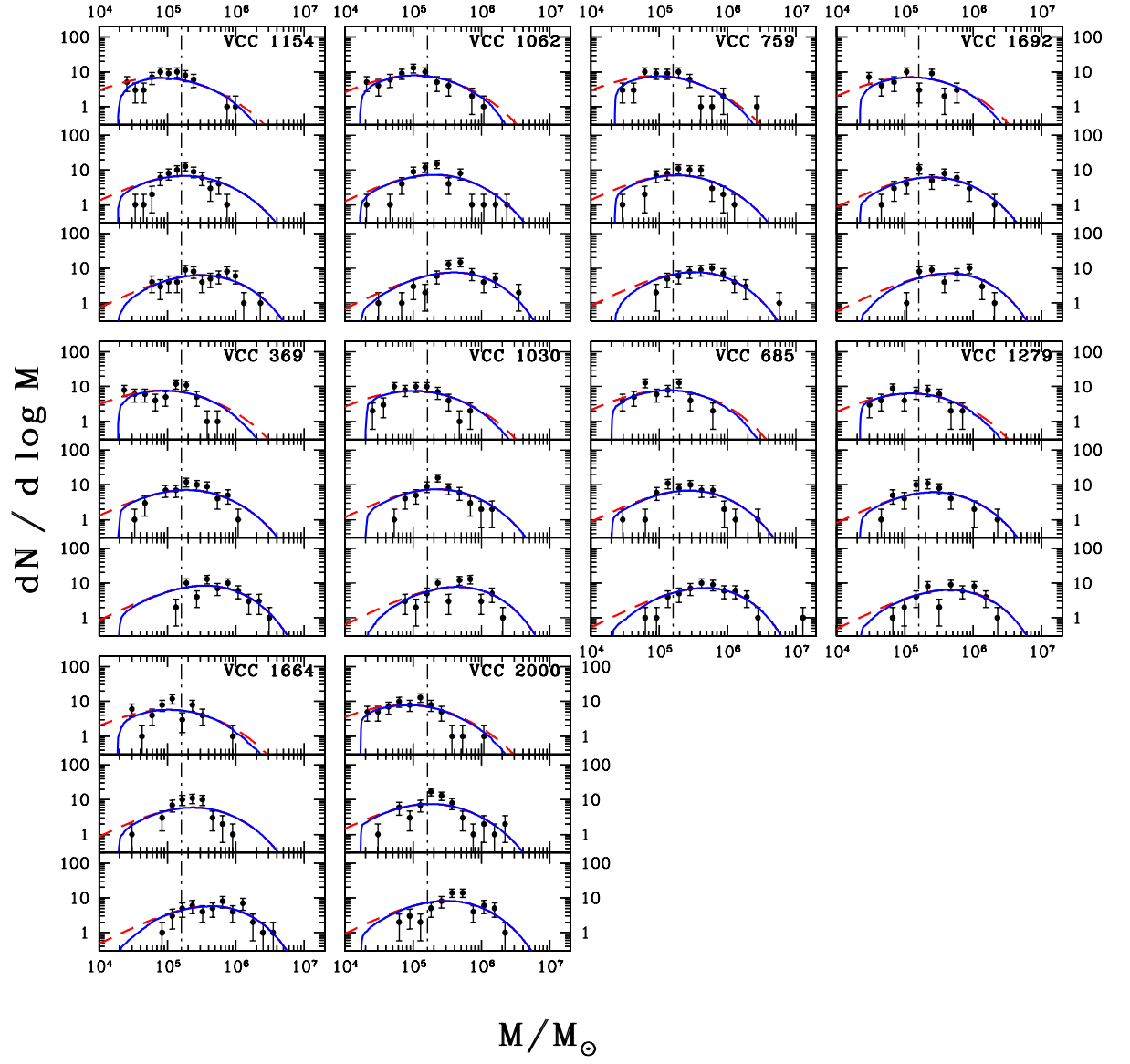


Figure 2.2: Continued.

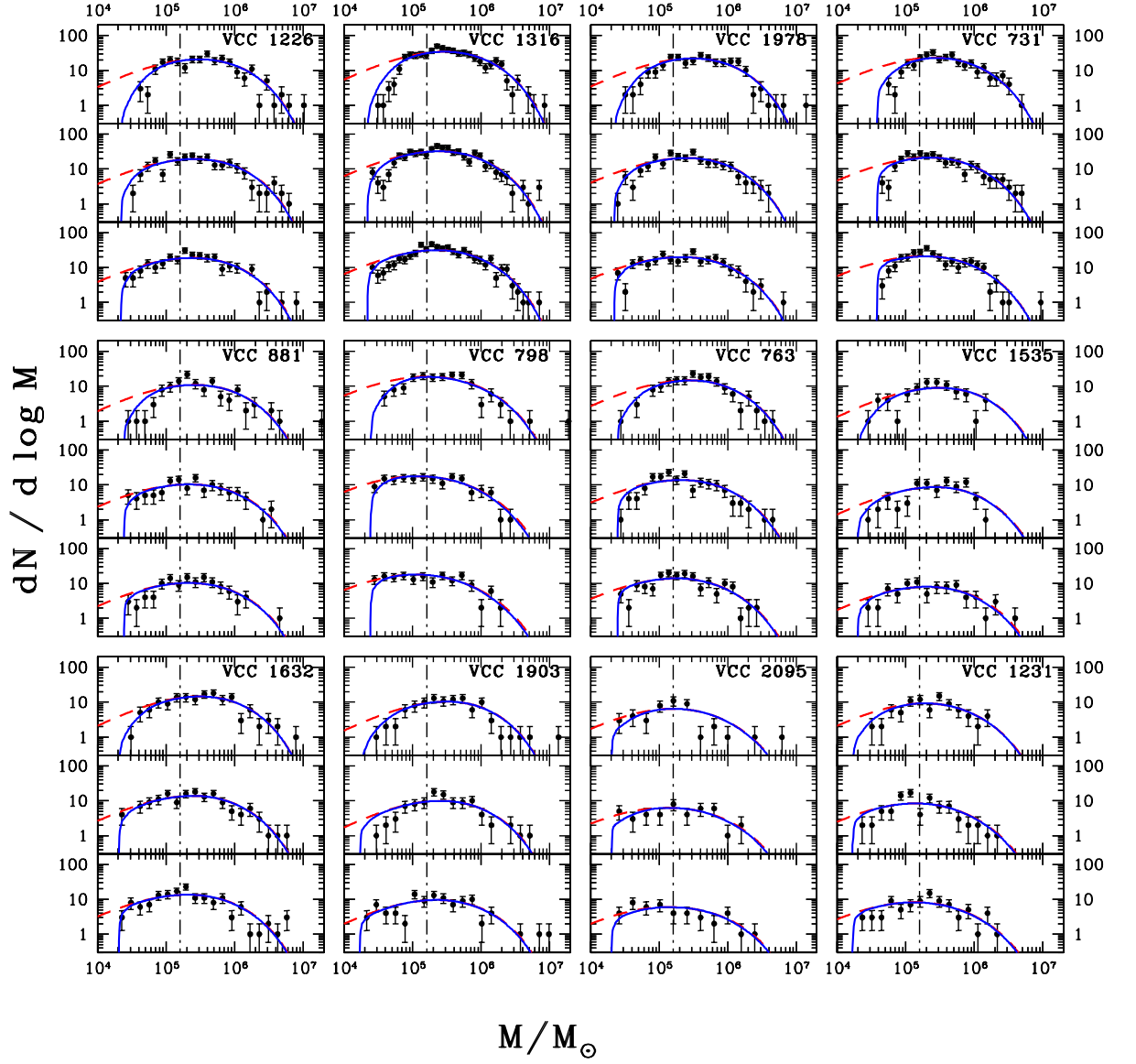


Figure 2.3: Observed mass function of the 22 ACSVCS galaxies selected in this sample, with evolved Schechter (1976) function (red dashed) and completeness degraded Schechter (1976) mass function (solid blue) applied to galactocentric position sorted data. Galaxies are ordered by descending B -band luminosity. Details of each plot are given in Table 2.1.

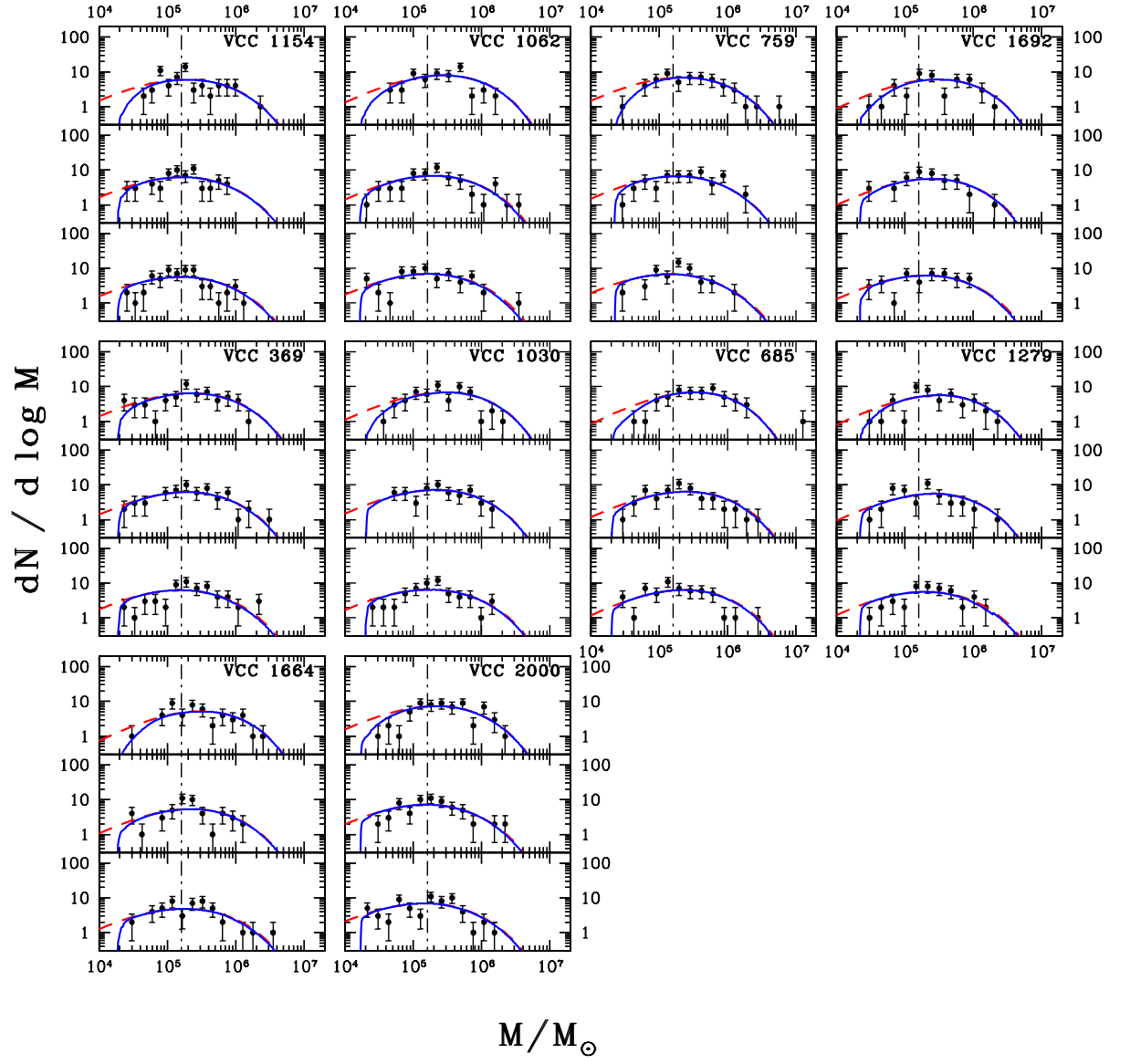


Figure 2.3: Continued.

Table 2.1: Details of the models displayed in Figures 2.2 and 2.3.

Name	$\log[L_{\text{gal}}]$ [L_{\odot}]	R_{eff} [kpc]	\mathcal{N}	GCSs sorted by ρ_h			$\log C$ [$M_{\odot}^{1/2} \text{pc}^{3/2} \text{Gyr}^{-1}$]	GCSs sorted by R_{gc}		
				$\log[\hat{\rho}_h]$ [$M_{\odot} \text{pc}^{-3}$]	$\log[M_{TO}]$ [M_{\odot}]	χ^2_{ν}		$\log[\hat{R}_{gc}]$ [kpc]	$\log[M_{TO}]$ [M_{\odot}]	χ^2_{ν}
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
VCC 1226	10.984	210.80	749	2.079	5.104	12.4	2.98	0.514	5.480	2.6
				2.874	5.432	9.1		0.788	5.416	1.7
				3.472	5.688	6.2		0.943	5.376	1.7
VCC 1316	10.878	155.40	1717	2.176	5.104	22.5	2.96	0.471	5.464	2.9
				2.847	5.408	14.2		0.752	5.384	2.8
				3.382	5.640	12.7		0.928	5.352	2.1
VCC 1978	10.791	98.14	787	2.207	5.072	9.0	2.90	0.447	5.488	1.2
				2.974	5.416	5.4		0.741	5.400	1.7
				3.561	5.656	7.4		0.912	5.360	1.7
VCC 731	10.742	110.60	881	2.195	5.056	14.8	2.88	0.549	5.480	2.3
				2.859	5.352	6.3		0.857	5.376	2.4
				3.451	5.600	4.3		1.043	5.288	2.6
VCC 881	10.736	226.80	360	1.908	5.032	3.8	2.98	0.525	5.400	1.8
				2.666	5.352	5.2		0.794	5.328	0.9
				3.358	5.640	2.9		0.946	5.328	1.3
VCC 798	10.725	65.32	497	1.668	4.896	4.5	2.83	0.482	5.240	1.3
				2.405	5.128	5.4		0.754	5.072	0.9
				3.164	5.440	4.8		0.931	5.048	1.5
VCC 763	10.675	154.80	498	1.991	5.024	10.8	2.90	0.457	5.448	1.5
				2.812	5.344	3.8		0.762	5.320	2.2
				3.361	5.616	4.9		0.943	5.248	2.7
VCC 1535	10.424	47.10	240	1.913	5.080	1.8	3.03	0.364	5.488	4.1
				2.708	5.416	1.3		0.644	5.424	2.6
				3.290	5.664	1.4		0.884	5.304	1.0
VCC 1632	10.387	44.97	447	2.245	5.056	4.8	2.88	0.328	5.504	1.2
				3.010	5.408	5.0		0.672	5.392	0.9
				3.592	5.648	4.5		0.882	5.320	1.9
VCC 1903	10.300	107.10	305	2.229	5.096	3.5	2.91	0.231	5.496	1.1
				2.956	5.424	2.0		0.606	5.400	1.7
				3.545	5.664	3.1		0.840	5.352	1.9
VCC 2095	10.196	19.38	131	1.816	4.904	4.7	2.79	0.281	5.216	2.6
				2.594	5.152	3.6		0.638	5.136	0.9
				3.358	5.488	1.0		0.816	5.120	0.7
VCC 1231	10.158	26.99	252	2.132	4.896	2.6	2.72	0.197	5.272	1.2
				2.837	5.200	2.6		0.593	5.144	1.9
				3.345	5.464	2.7		0.819	5.120	1.4
VCC 1154	10.118	28.90	190	1.941	4.928	1.5	2.80	0.229	5.280	1.6
				2.799	5.248	3.0		0.545	5.216	1.3
				3.300	5.496	1.0		0.799	5.208	1.8
VCC 1062	10.028	17.81	176	2.027	5.016	1.1	2.94	0.268	5.440	1.8
				2.634	5.304	3.3		0.522	5.288	1.2
				3.299	5.592	2.3		0.794	5.224	1.9
VCC 759	10.011	34.13	165	1.957	5.008	1.9	2.87	0.120	5.368	0.4
				2.664	5.256	2.1		0.483	5.256	1.1
				3.241	5.512	0.8		0.804	5.160	2.0
VCC 1692	9.980	13.28	128	1.984	5.104	2.0	3.04	0.195	5.464	1.5
				2.681	5.392	1.1		0.552	5.384	1.0
				3.197	5.640	2.7		0.873	5.304	2.2
VCC 369	9.949	31.00	177	1.846	4.960	3.3	2.91	0.307	5.360	1.8
				2.661	5.280	2.2		0.599	5.288	1.0
				3.217	5.544	2.1		0.802	5.184	2.1
VCC 1030	9.944	17.23	170	2.027	5.008	1.9	2.90	0.200	5.472	2.1
				2.797	5.344	2.6		0.550	5.336	0.7
				3.513	5.632	2.8		0.853	5.280	1.5
VCC 685	9.882	12.37	165	1.986	5.136	2.0	3.07	0.226	5.520	0.7
				2.730	5.440	2.1		0.589	5.400	0.9
				3.262	5.672	0.9		0.851	5.368	2.0
VCC 1279	9.851	11.70	136	1.946	5.080	0.8	3.05	0.170	5.504	1.5
				2.698	5.424	1.7		0.604	5.424	1.4
				3.259	5.656	1.8		0.907	5.296	1.4
VCC 1664	9.838	16.13	142	2.073	5.016	2.8	2.92	0.055	5.472	1.1
				2.823	5.376	2.5		0.438	5.344	2.6
				3.401	5.632	0.7		0.773	5.248	1.0
VCC 2000	9.832	10.61	192	2.062	4.920	2.2	2.79	0.128	5.344	2.2
				2.822	5.248	3.1		0.515	5.192	1.2
				3.355	5.488	2.5		0.765	5.176	2.1

Key to columns: (1): VCC catalogue number (Jordán et al. 2009), (2): Log of galaxy B -band luminosity (Mei et al. 2007), (3): Galaxy effective radius (Mei et al. 2007), (4): Number of clusters in sub-population, (5): Log of median half-light density of each division of density sortings, (6): Log turnover of GCMF model applied to this density division, (7): Reduced chi-squared of GCMF model fit in this density division, (8): Best-fit value Log C , (9): Log of median galactocentric distance of radially-sorted divisions, (10): Log turnover of GCMF model applied to this radial division, (11): Reduced chi-squared of GCMF model fit in this radial division.

Table 2.2: Statistics of 22 galaxy fits for different values of M_{sch} .

M_{sch} (M_{\odot})	β_{MF}	\hat{C}	$IQR(C)$	\bar{C}	$\sigma(C)$
(1)	(2)	(3)	(4)	(5)	(6)
0.5×10^6	2	3.35	0.25	3.35	0.144
0.8×10^6	2	3.15	0.15	3.16	0.111
1.0×10^6	2	3.10	0.20	3.09	0.114
1.5×10^6	2	3.00	0.10	3.00	0.091
2.0×10^6	2	3.00	0.15	2.96	0.091
2.5×10^6	2	2.91	0.11	2.91	0.092
3.0×10^6	2	2.90	0.10	2.90	0.084
3.5×10^6	2	2.90	0.05	2.88	0.082

(1): Exponential cut-off mass, (2): Exponent of young cluster mass function power-law, (3): Median value of $\log C$, (4): Inter quartile range of $\log C$, (5): Mean value of $\log C$, (6): Standard deviation of $\log C$.

very large r_h , with some in excess of 10 pc. Similarly, the lower limit of 0.75 pc imposed by Jordán et al. (2009) in their catalogues will lower the blue curve at the lower mass end of each bottom panel — however this effect is much less obvious due to other incompleteness effects, such as the intrinsic faintness of low-mass clusters. The vertical dot-dashed line denotes the position of the canonical Milky Way GCMF turnover at $M \simeq 1.6 \times 10^5 M_{\odot}$. Using this as a visual aid, it is plain to see the turnover of the GCMF decrease towards lower cluster densities, yet remain roughly constant in different R_{gc} ranges; reasserting the well known weak dependence of the GCMF peak on the galactocentric positions of its GCs (e.g., Harris, Harris & McLaughlin 1998; Barmby, Huchra & Brodie 2001; Vesperini et al. 2003; Jordán et al. 2007; Chandar, Fall & McLaughlin 2007; McLaughlin & Fall 2008).

Figure 2.3 displays the same equations (2.9, solid curve) and (2.9 with $f(M_{c,i}, \rho_{h,i}, I_{b,i}) = 1$ for all $M_{c,i}$, $\rho_{h,i}$, and $I_{b,i}$, dashed curve), but applied to the GC data sorted by R_{gc} (i.e. the top panel is computed with the densities of one third of

total clusters at the smallest projected galactocentric distances, etc.). Additionally, the models were applied with the value of C minimised by the fits in Figure 2.2. Generally, the data are better fitted by the models when sorted by R_{gc} as in Figure 2.3 than by ρ_h , as in Figure 2.2 below. This is a result of ρ_h being a weak function of R_{gc} , namely the large scatter in ρ_h vs R_{gc} as displayed for some example galaxies in Figure 2.6 (and also as noted by McLaughlin & Fall 2008; Chandar, Fall & McLaughlin 2007; Goudfrooij 2012 for the Milky Way, M104, and NGC1316). As a consequence of this scatter, selecting GCs based on their R_{gc} is essentially equivalent to selecting GCs randomly in terms of ρ_h , and thus typically the average ρ_h of any sub-population of GCs sorted by R_{gc} will be the same. Thus, any dependence the GCMF may have on cluster density is obscured in each of the three panels. When the sub-populations are sorted by ρ_h , the models need to accurately account for any GCMF density-dependence in order to achieve a good fit. The actual physical situation is likely to be much more complicated than that which is encapsulated with these simple single parameter models, and hence achieving a good fit to ρ_h sorted data is much more difficult. Consequently, the models tend to fit R_{gc} -sorted data better than ρ_h -sorted data.

Figure 2.4 was then constructed from the data in Table 2.1, displaying the dependence of M_{TO} on both median half-light density ($\hat{\rho}_h$, top panel) and median galactocentric position (\hat{R}_{gc} , bottom panel). Data points from each division of the fits (black points) and from observational data (red points), in addition to results for the Milky Way from the analogous study by McLaughlin & Fall (2008) (open stars) are plotted in both panels. In the top panel, equation (2.8) is plotted using the median value of C minimised for models fits to all 22 ACSVCS galaxies, and $M_c = 2.5 \times 10^6 M_\odot$. Least-squares fitting in the top panel gives a logarithmic gradient of 0.491 ± 0.019 , and the Spearman rank coefficient for these data is 0.909 with a p -value of < 0.00001 . As mentioned in section (2.3), if $C\rho_h^{1/2}t \ll M_{sch}$, then $M_{TO} \simeq C\rho_h^{1/2}t/(\beta_{MF} - 1)$; and so using $C \simeq 800 M_\odot^{1/2} \text{pc}^{3/2} \text{Gyr}^{-1}$, $\rho_h \simeq 200 M_\odot \text{pc}^{-3}$, $t = 13 \text{ Gyr}$, and $M_{sch} = 2.5 \times 10^6 M_\odot$, then $C\rho_h^{1/2}t/M_{sch} \simeq 0.06$. Thus, these statistics are consistent with the expectation that $M_{TO} \propto \rho_h^{1/2}$. Note that the use of the median of ρ_h from each division of each galaxy in equation (2.8) is only giving an estimate of M_{TO} , the appropriate average

of ρ_h to get back the actual turnover-mass is likely much more complicated. In the bottom panel, the maxima of the model fits and median half-light density are plotted against median R_{gc} for each division of R_{gc} -sorted data in all 22 galaxies. Since roughly again $M_{TO} \propto \rho_h^{1/2}$, and for a singular isothermal sphere $\rho_J \propto r_{gc}^{-2}$ (cf. equation 2.1), then roughly one would expect $M_{TO} \propto R_{gc}^{-1}$. A line displaying this expected gradient between M_{TO} and R_{gc} is displayed (note that the intercept is arbitrary, this line is for display purposes only). Least-squares fitting in the bottom panel gives a logarithmic gradient of -0.210 ± 0.053 , and the Spearman rank coefficient for these data is -0.335 with a p -value of 0.0272. These statistics are much less compatible with a $M_{TO} \propto R_{gc}^{-1}$ relation. However, R_{gc} is in projection, tides may not actually be set at pericentre, and although most clusters may be expected to be near apocentre, there will be a large spread in orbital phase across the entire GC population. Thus the observed M_{TO} vs R_{gc} correlation will likely be different from what is expected, but probably not to the extent as in Figure 2.4. This is because although tides are probably not set at pericentre, studies indicate that the effective galactocentric radius corresponding to tidal limitation is not very far from pericentre, either (Webb et al. 2013). Furthermore, the data from the Milky Way, which are not projected, fit amongst the rest of the data without standing apart in any significant way. Therefore, it seems unlikely that the expected $M_{TO} \propto R_{gc}^{-1}$ is present but has been concealed by these effects. This is discussed further in section 2.5.

The simultaneous dependence of M_{TO} on $\hat{\rho}_h$ plus weak dependence on \hat{R}_{gc} in Figure 2.4 is made possible by scatter in ρ_h vs R_{gc} , as displayed in Figure 2.6 for 4 sample galaxies. Previous attempts to explain this absence of expected dependence of M_{TO} on R_{gc} have relied on velocity anisotropy, namely increasingly eccentric orbits with larger galactocentric radii such that the average pericentre of cluster orbits at all galactocentric radii is roughly constant. Thus, the tidal density of all clusters would be similar, and consequently so would the mass-loss rate, and therefore the same GCMF would be produced at all galactocentric radii. However, analyses of GCS velocity distributions find much less anisotropy than is required (e.g., Vesperini et al. 2003, Fall & Zhang 2001). Thus, the same as in the Milky Way (McLaughlin & Fall 2008) and

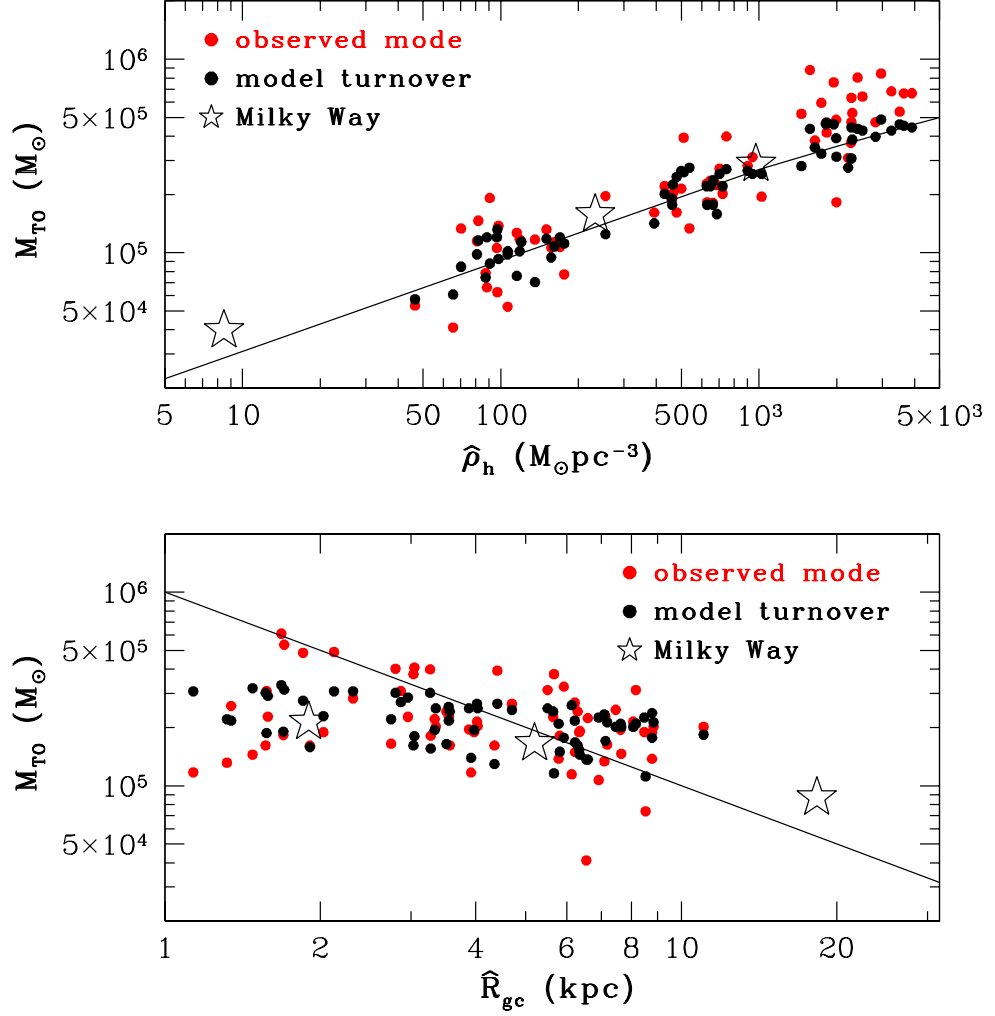


Figure 2.4: Top panel: M_{TO} vs $\hat{\rho}_h$ for density fitted models, with equation (2.8) plotted. Bottom panel: M_{TO} vs \hat{R}_{gc} for galactocentric distance fitted models with the expected slope of $M_{TO} \propto R_{gc}^{-1}$ plotted (see text for details). Modes of each sub-population's observational data are in red, maxima of models are in black.

in M104 (Chandar, Fall & McLaughlin 2007) it appears that it is scatter in ρ_h vs R_{gc} that allows evaporation dominated mass-loss to carve a radially invariant GCMF that depends on internal cluster density out of a power-law CIMF.

The models given by equation (2.9) were computed using a range of values for the parameter M_{sch} , and subsequently compared. These results can be found in Table 2.2 (note that models presented in Figures 2.2 and 2.3 were fixed with $M_{sch} = 2.5 \times 10^6 M_\odot$, as discussed in section 2.3. On the whole, the optimum value of M_{sch} for each galaxy seems to reflect the individual characteristics of its GCMF. For example, the relative dip in the number of massive clusters in the top panel of Figure 2.2 for VCC 1316 causes the model to generate a lower χ^2 if it is shifted towards lower masses — thus the value of M_{sch} corresponding to the lowest χ^2 for this galaxy was actually $1.5 \times 10^6 M_\odot$; even though based on increasing M_{sch} with galaxy luminosity as found by Jordán et al. (2007), one might expect better fits with higher M_{sch} for massive galaxies. This is because a larger value for M_{sch} extends the models further into the high cluster mass regime, and as larger galaxies generally have broader cluster mass functions, this would be expected to produce a better fit. For the same reasons, a smaller M_{sch} would be expected to fit smaller galaxies better. However, whilst this expectation might be accurate for a few of the smoother GCMFs (e.g., VCC 1903, 1231, 1030, 2000 and 1279 each have optimum M_{sch} values of $1.5 \times 10^6 M_\odot$), overall, $M_{sch} = 2.5 \times 10^6 M_\odot$ gave the smallest total χ^2_ν summed over all galaxies, and so this value was adopted throughout. The overall effect of increasing M_c was to decrease C (Table 2.2, columns 3 and 5) and reduce the spread in the values of C (columns 4 and 6). This is because for most of the galaxies in this sample of 22, a higher M_{sch} causes the models to overpredict the number of high-mass clusters. Hence, during the process of χ^2 minimisation, the model is pulled towards lower masses to better fit the data, resulting in a smaller M_{TO} . Since for $C\rho_h^{1/2}t/M_{sch} \ll 1$, $M_{TO} \propto C\rho_h^{1/2}$, this means that C must decrease for fixed ρ_h . Furthermore, the results show that as M_{sch} tends towards infinity (i.e. with possible cluster masses extending up to arbitrarily high values, or equivalently equation (2.6) with the exponential removed), the dependence of $\log[M_{TO}]$ on $\log[\rho_h]$ tends towards 0.5 exactly, in perfect agreement with the theory (as can easily be

obtained by differentiating equation (2.6) with the exponential term removed, and solving for the maximum in M). Thus, changing M_{sch} only changes the numerical value of C slightly (a change in M_{sch} of a factor of 7 results in a change in C of a factor less than 3). Assuming an average half-mass density of $\bar{\rho}_h = 200 \text{ M}_\odot \text{pc}^{-3}$, the ratio $C\rho_h^{1/2}t/M_{sch}$ for the top row of Table 2.2 evaluates to 0.82, descending down to 0.04 for the bottom row. When $C\rho_h^{1/2}t/M_{sch} \ll 1$, $M_{TO} \simeq C\rho_h^{1/2}t/(\beta_{MF} - 1)$, and thus over the majority of the parameter space investigated, C and β_{MF} are degenerate. Thus the CIMF power-law index was kept fixed at $\beta_{MF} = 2$ as this provides good fits to the mass functions of young cluster systems, as observed in mergers such as the Antennae (e.g., Whitmore & Schweizer 1995).

As mentioned in section 2.2, the results presented here are all derived from LP -band data. Fits were also made to the W -band data, to check for consistency. As such, W -band GCMF peaks were found to be typically ~ 0.05 dex lower than those inferred from LP -band. This is because $R_{h,W}$ measurements were typically $\sim 0.074\%$ smaller than $R_{h,LP}$ on average (Jordán et al. 2005), and roughly, $\Delta M_{TO} \propto \Delta \rho_h^{1/2} \propto \Delta r_h^{-3/2}$ (cluster masses were not included in this estimate because when averaged over a large number of clusters, the W and LP masses actually agree very well and thus had a negligible contribution to this effect), thus $-1.5 \log[1 - 0.074] \approx 0.05$ dex. Figure 2.5 displays the difference between W - and LP -band turnover-masses (derived using density sorted sub-populations) as a fraction of their average for each division for all 22 ACSVCS galaxies. Generally, the data are in good agreement, with no obvious systematic trends. Additionally, the tendency of data points from the same data division to cluster together (i.e. lower density division turnovers are all below $1.4 \times 10^5 \text{M}_\odot$, intermediate density division turnovers are around $2.4 \times 10^5 \text{M}_\odot$, and higher density division turnovers are around $4.2 \times 10^5 \text{M}_\odot$) is a convenient demonstration that the turnover-mass of 22 galaxies spanning a wide range of properties is approximately the same, even though each galaxy has a unique GCS.

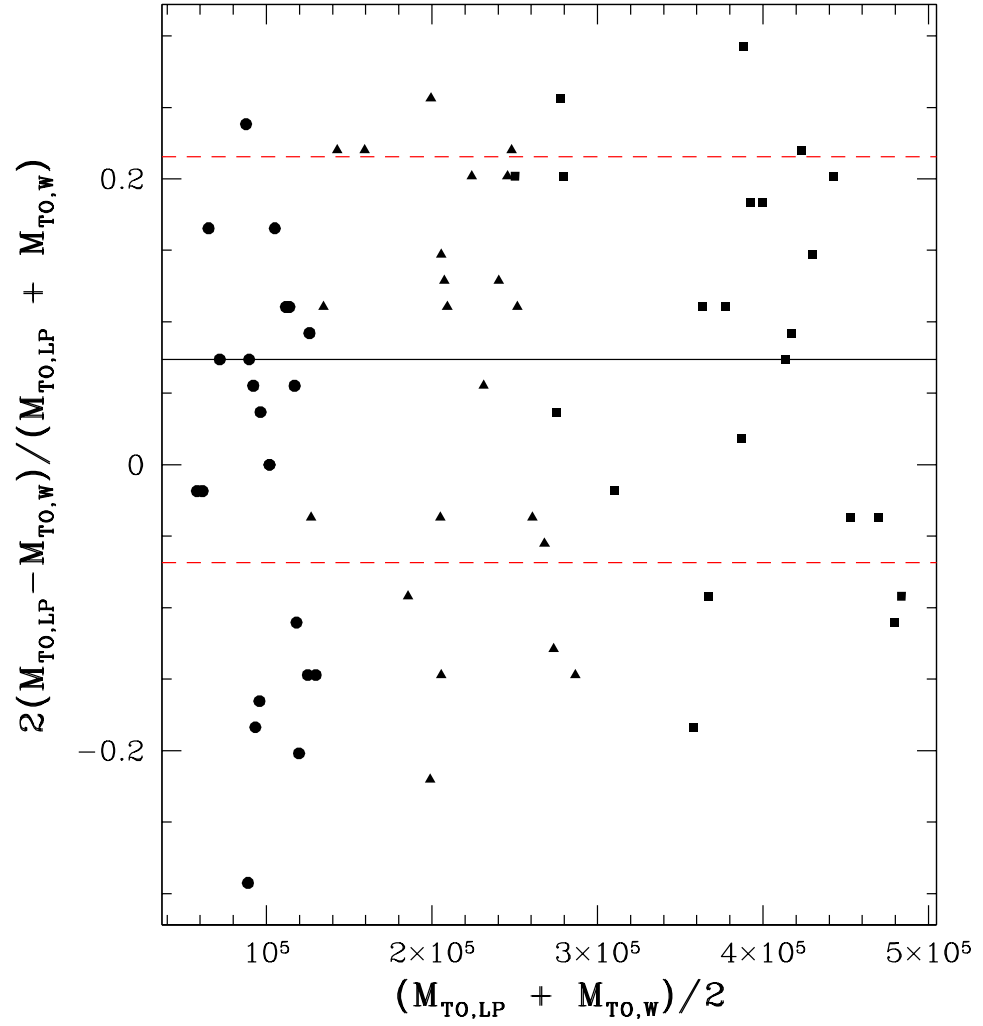


Figure 2.5: Difference between $M_{TO,W}$ and $M_{TO,LP}$ as a fraction of their average value against the average. Circles denote GCMF turnovers of low-density divisions, triangles denote GCMF turnovers of mid-density divisions, and squares denote GCMF turnovers of high-density divisions. The solid line denotes the median of all turnovers, and the dashed lines correspond to 1 standard deviation above and below this median.

2.5 Discussion

Inspection of the GCMF model fits in Figures 2.2 and 2.3 clearly shows that in general better fits are obtained when fitting to R_{gc} -sorted data rather than ρ_h -sorted data (cf. columns 7 and 11 in Table 2.1), as discussed in section 2.4. This indicates some additional GCMF dependence on internal GC density other than what is encapsulated in the GCMF models given by equation (2.9). When these densities are shuffled up (as when GCMF models are plotted with radially binned data, due to the lack of any correlation in cluster density vs galactocentric position, cf. Figure 2.6), the distribution of GC densities in each radial division are very similar, and consequently any GCMF dependence on GC internal density is no longer apparent, allowing better fits. In other words, these models capture GCMF dependence on ρ_h primarily, rather than R_{gc} . Even so, on the whole the fits provided by equation (2.9) do an impressive job describing evolved GCMFs, given the simplicity of the models.

Figure 2.2 clearly indicates that the GCMF evolution models fit less well for VCC 1226 and VCC 1316 (i.e. in the most massive galaxies); tending to predict GCMF widths in excess of observations, even after correcting for data incompleteness. There are several possible candidates/contributors to this effect: perhaps the most obvious is that the CIMF is not well described by a Schechter (1976) model with $\beta_{MF} \simeq 2$. Although young cluster systems observed today in active mergers (such as NGC 4038/4039, and the Antennae galaxies; Whitmore & Schweizer 1995) are well described by a Schechter (1976) CIMF, and these young clusters may well resemble GCs of today after 13 Gyr of dynamical evolution, this is not to say that the GCs of today had identical origins. Additionally, as these two galaxies are amongst the most massive in Virgo (in fact, VCC 1316/M87 is the most massive, and VCC 1226/M49 is the most luminous), in keeping with the view of hierarchical formation, they will both have very complicated merger histories, and thus it should not come as much of a surprise that a very simple single parameter model struggles to fit their GCSs properly.

Alternatively the problems with fitting VCC 1226 and VCC 1316 GCMFs could be the effect of one or more other destruction mechanisms not considered here, such as

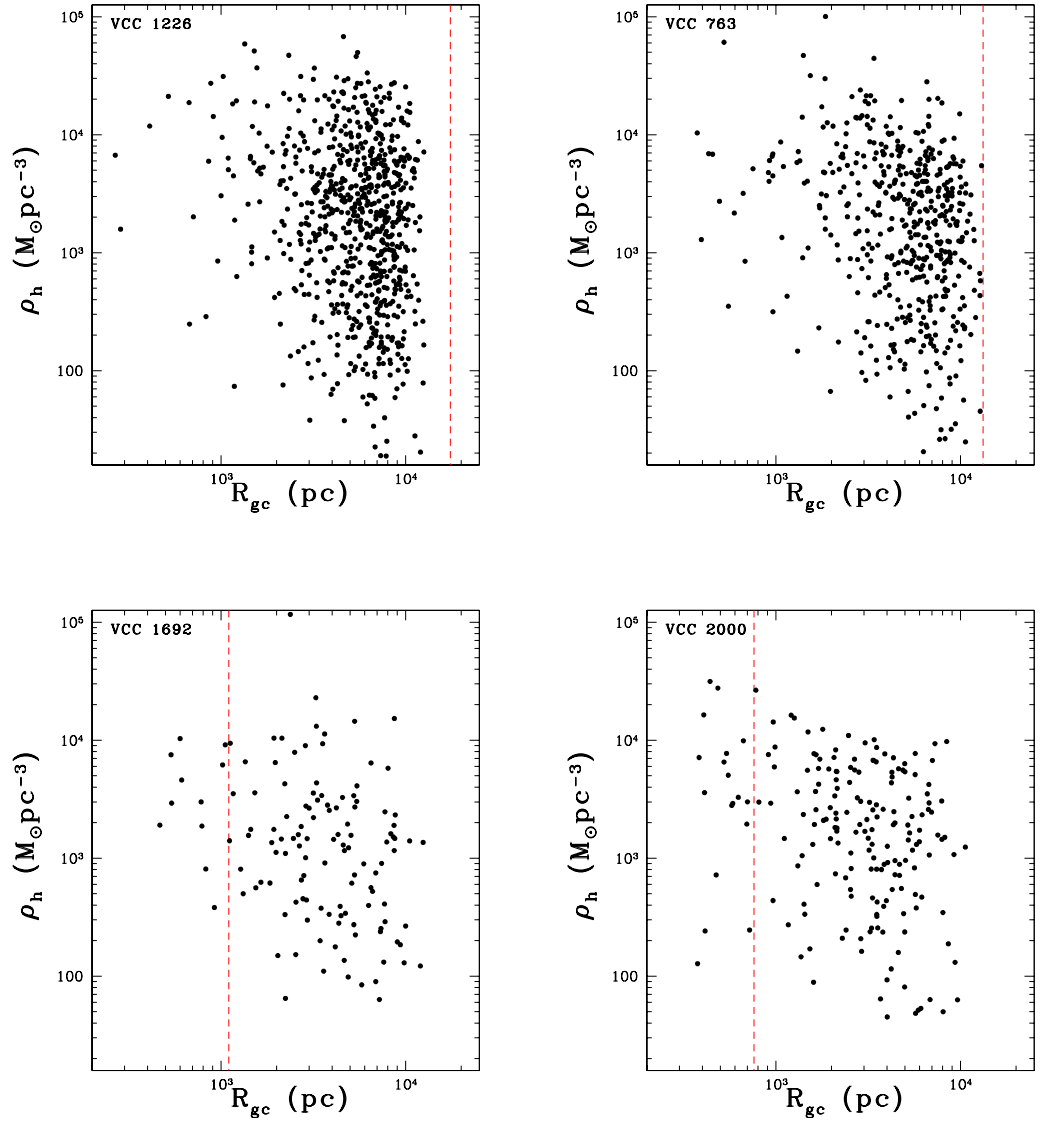


Figure 2.6: Internal cluster half-light density against galactocentric position. The red dashed line denotes the host galaxy's effective radius. These four plots are from galaxies at roughly even spacing from brightest to dimmest in B -band galaxy luminosity over the sample of 22 ACSVCS galaxies.

tidal shocking. This seems the most plausible explanation, given better fits with the same value of C when GCs are binned radially rather than by GC density as described earlier, since tidal shocking is in some cases expected to be as important as evapo-

ration (e.g., Gnedin, Lee & Ostriker 1999), and depends on GC density (see section 1.4.6). Treating for shocks in addition to evaporation is difficult, as shock destruction rates depend on fine details of individual cluster orbits and galaxy potentials, not just on mean tidal fields (e.g., Fall & Zhang 2001), and so is not attempted in this investigation. Chapters 3 and 5 seek to develop more advanced models than those used here, capable of including much more physics, including tidal shocks. Alternatively, this effect could simply be reflecting the approximations and assumptions made in this treatment, such as approximating cluster evaporation rates as constant (which has the implicit assumption that the galactic potential is static and time-independent). Alternatively, it could be the assumption that all of these clusters are tidally limited (e.g., see Gieles, Baumgardt & Heggie 2010; Baumgardt et al. 2010), or that they are not all 13Gyr old. However, the main point of the fits is that they clearly demonstrate that GCMF peak mass (M_{TO}) depends on the densities of clusters being used in the fitting, with the peak shifting to higher masses for more dense clusters. When these densities are shuffled up, as when the binning thirds are based on R_{gc} , these details are averaged out, as can be seen in Figure 2.3.

Figure 2.4 shows the turnover-mass (M_{TO}) vs median internal cluster half-light density ($\hat{\rho}_h$) given in Table 2.1 for each division of all 22 ACSVCS galaxies' GCS in the top panel, with equation (2.8) plotted. The lower panel shows the tabulated values of M_{TO} vs median galactocentric position \hat{R}_{gc} from Table 2.1, with the expected dependence plotted ($M_{TO} \propto \hat{R}_{gc}^{-1}$) as discussed in section 2.4); unsurprisingly, the results generated by the models calculated from the same equation fit this line very closely — more relevant is that the mode of the observed GCMFs clearly follow the trend of this line also, albeit with more scatter. This demonstrates that the simple treatment of 13 Gyr of evaporation dominated GCMF evolution provides a good match with observations. The tendency of observed data points to sit above the curve at high $\hat{\rho}_h$ is due to model M_{TO} being lower than the mode of the data in the high-density divisions. This in turn is because of a common feature in the low-density divisions of the most massive galaxies in the sample, namely the unexpected low number of massive, low-density clusters, which causes a smaller χ^2 for fits favouring lower masses.

This effect is most pronounced for VCC 1316, VCC 1226, VCC 1978 and VCC 731. The bottom panel of Figure 2.4 has the expected dependence of M_{TO} on \hat{R}_{gc} plotted. As discussed in section 2.4, M_{TO} scales roughly as $\rho_h^{1/2}$, thus, as the density of a tidally limited cluster in a singular isothermal sphere is expected to scale as $\rho_h \propto V_c^2/r_{gc}^2$, where V_c is the galaxy circular speed, we would expect that roughly $M_{TO} \propto V_c/R_{gc}$. Clearly the intercept of this line will depend on V_c , which will be different for each galaxy. Even so, neither the observational data nor the model generated data points fit the general trend of this line. The reason for this is displayed in Figure 2.6. These four panels show ρ_h as a function of R_{gc} for four galaxies in roughly even steps from brightest to dimmest in the sample of 22, each with their effective radius denoted with a dashed line. The sudden break in the data at around $R_{gc} \approx 1.2 \times 10^4$ pc is at 150 arcseconds at the distance of Virgo, the size of the ACSVCS field of view. Each panel (plus the other 18 galaxies not displayed) show only scatter in ρ_h as a function of R_{gc} , rather than the roughly expected $\rho_h \propto R_{gc}^{-2}$. With this scatter, it is possible for M_{TO} to vary with ρ_h and yet be independent of R_{gc} . This same scatter in ρ_h vs R_{gc} and weak dependence of M_{TO} on R_{gc} and yet strong dependence of M_{TO} on ρ_h was also found by McLaughlin & Fall (2008) in the Milky Way, by Chandar, Fall & McLaughlin (2007) in M104, and by Goudfrooij (2012) in NGC1316.

Aside from being a useful diagnostic for biases, Figure 2.5 also demonstrates that the M_{TO} values from each density division in both W and LP tend to bunch together and form distinct groups — indicating that the densities of GCs of each division in all 22 galaxies that went into producing these values are all roughly the same on average also. Figure 2.7 also demonstrates this effect. The top panel shows each galaxy’s GCS median density, and the bottom panel shows turnover-mass vs the galaxy’s luminosity in this sample of 22. Over a factor of 15 in galaxy luminosity, the medians of cluster densities in each galaxy vary by only a factor of about 4. This is because large galaxies all tend to have roughly the same ambient densities on average. Hence, galactic tidal fields all tend to be similar, and as these tides set the densities of each galaxy’s globular clusters (provided they are all tidally limited), there is a relatively limited range in the densities of GCs. Since roughly $M_{TO} \propto \rho_h^{1/2}$, this relatively limited range in GC

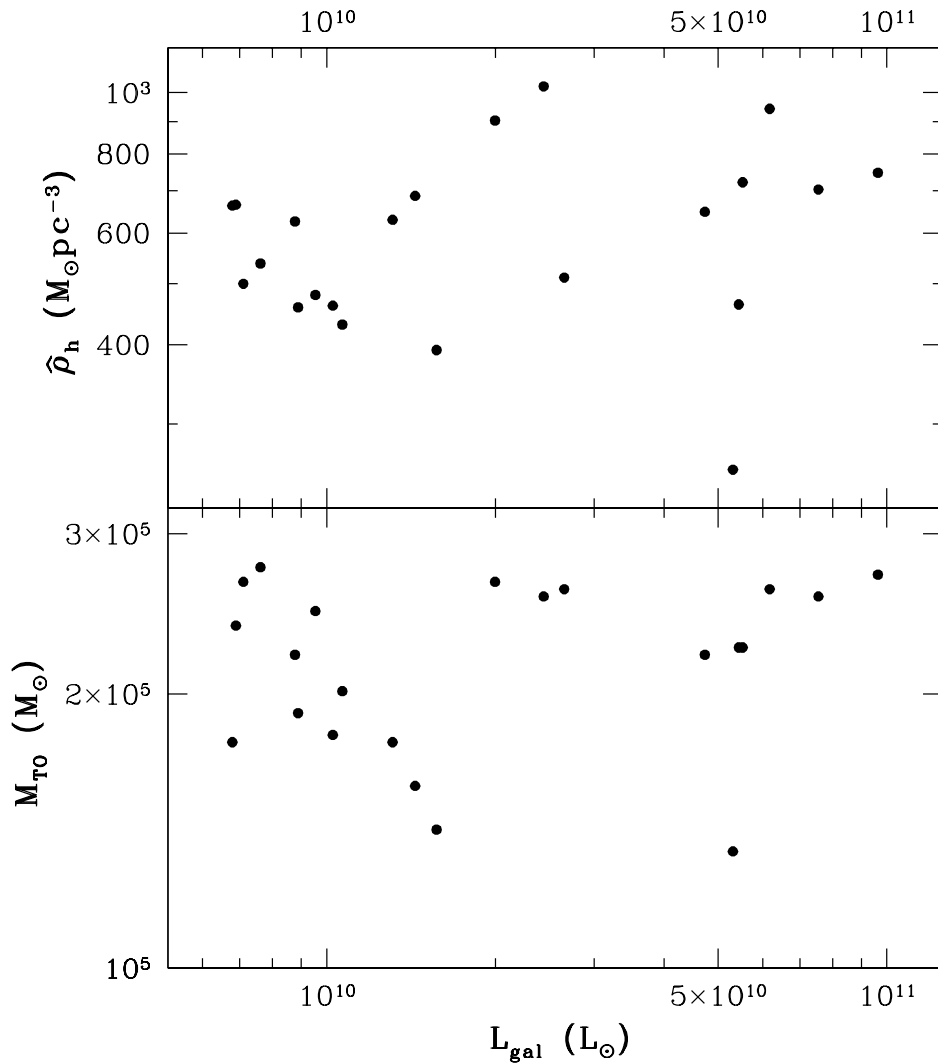


Figure 2.7: Top panel: median density of globular clusters against host galaxy luminosity. Bottom panel: GCMF turnover-mass against host galaxy luminosity.

density translates to an even more limited range in M_{TO} (only a factor of about 3 in the bottom panel of Figure 2.7). As a result, most galaxies have very similar values for M_{TO} , creating the impression that GCMF peaks are ‘universal’.

The main result of this Chapter is the average value of $C_{virgo} = 810 \pm 170$ calculated for $M_{sch} = 2.5 \times 10^6 M_\odot$ given in Table 2.2. This value is in excellent agreement with those calculated by Jordán et al. (2007) in Virgo, McLaughlin & Fall (2008) in

the Milky Way, Chandar, Fall & McLaughlin (2007) in M104, and Goudfrooij (2012) in NGC1316 ($C_J = 840$, $C_{MF} = 1100$, $C_{CFM} = 560$, $C_G = 875$), which is a good result considering the simplicity of these models and that the parameter C was unconstrained during the fitting in each of these studies. Additionally, this value is in good agreement with theoretical predictions about the lifetimes of tidally limited clusters undergoing two-body relaxation driven evaporation. Using $C = 810$, $m = 0.7$, $\lambda = 0.4$, and assuming $N = 10^5$, equation (2.4) gives $\xi \simeq 0.078$, meaning that this fraction of mass lost per relaxation time predicts that clusters are expected on average to survive for about another 13 relaxation times before being destroyed, in good agreement with other studies (e.g., Hénon 1961; Spitzer & Chevalier 1973; Spitzer & Shull 1975; Fall & Rees 1977; Caputo & Castellani 1984; Spitzer 1987; Aguilar, Hut & Ostriker 1988; Chernoff & Weinberg 1990; Gnedin & Ostriker 1997; Murali & Weinberg 1997; Gnedin, Lee & Ostriker 1999; Jordán et al. 2007; Chandar, Fall & McLaughlin 2007; McLaughlin & Fall 2008; Goudfrooij 2012).

The lack of any correlation in ρ_h vs R_{gc} for all Virgo GCSs (cf. Figure 2.6) means that any sub-population of GCs selected by R_{gc} sorting is essentially equivalent to building sub-populations with GCs selected randomly in terms of ρ_h . Consequently, the median GC densities of each sub-population, $\hat{\rho}_h$, tend to be very similar. Thus, given the systematic dependence of M_{TO} on $\hat{\rho}_h$, similar median GC densities lead to similar turnover-masses. Thus, M_{TO} is largely independent of \hat{R}_{gc} , i.e., the GCMF is radially invariant. When combined with the result that M_{TO} depends on $\hat{\rho}_h$ in a way that closely matches that expected for evaporation, this Chapter demonstrates that evaporation dominated mass-loss can turn an initial power-law CIMF with $\beta_{MF} \simeq 2$ into a GCMF as observed today in the Milky Way and many extant galaxies, adding more weight to the growing evidence that GCMF evolution is indeed dominated by evaporation.

The next step is to develop more sophisticated models including more physics, and with some of the assumptions made here relaxed in an attempt to reconcile the evolution of GCMF shape and radial invariance in all galaxies, including very massive ones. To this end, Chapter 3 next describes the general procedure for setting up

an initial GCS with specifiable initial mass function, initial spatial distribution, and initial kinematic distribution in a wide variety of galaxy models. Chapter 4 then demonstrates some of these simulated initial GCS models. Chapter 5 then describes the procedure for evolving simulated initial GCS through time, to facilitate comparisons with observations. The code applying this procedure for evolving simulated GCSs is then described, and demonstrations of its application are given in Chapter 6.

3 Initial Set-up of Simulated GCSs

3.1 Introduction

In this Chapter, the intention is to derive the properties of a tracer population orbiting in the gravitational potential of a host galaxy in a general way. This initial tracer population must have specifiable tracer number density, tracer velocity, and tracer mass distributions. To this end, quasi-separable distribution functions are employed (Gerhard 1991). Once the initial GCS properties have been obtained, Monte Carlo sampling is used to assign initial galactocentric positions, masses, orbital energies, and orbital angular momenta to each object of a desired population. With these initial data, all other orbital properties can be calculated, such as instantaneous radial, azimuthal, and polar velocities, orbital pericentres and apocentres, etc. With this information on every object, it is then possible to evolve these objects in time by applying dynamical effects such as evaporation, tidal shocks, stellar evolution, internal evolution, and dynamical friction, as described in Chapter 5.

3.2 Quasi-Separable Distribution Functions

For the most part, only a limited number of anisotropy profiles such as Osipkov-Merritt (Osipkov 1979; Merritt 1985) have been utilised when investigating GCS evolution. Furthermore, only a small variety of host galaxy potentials (almost exclusively the singular isothermal sphere) and GCS density profiles (usually a power-law GCS density profile) have been used. The reasoning behind evolving GCSs with velocity anisotropy is that the correct velocity anisotropy profile may provide a natural explanation to the observed radial invariance of the GCMF. For example, Vesperini et al. (2003) used Osipkov-Merritt models to introduce radial anisotropy into their initial GCS. The initial GCS was then evolved for a Hubble time, and the final mass function and velocity profile were compared to the GCS of M87. Although they recovered the radial

invariance of the GCMF, they concluded that the amount of radial velocity anisotropy required was incompatible with observations. By also varying the host galaxy potential and GCS density profile, this tantalisingly close result may be able to explain the radial invariance of the GCMF, and thus deserves closer attention.

In order to more fully understand the impact of velocity anisotropy on GCS evolution, a greater variety of initial velocity anisotropy profiles should be employed in an otherwise identical distribution. In general, this is quite a complicated problem, as the anisotropy profile is controlled by the angular momentum dependence of the distribution function, which also determines the GCS spatial distribution. Thus changing one without affecting the other is a difficult problem. In order to solve this problem, quasi-separable distribution functions (Gerhard 1991) are utilised (e.g. Gerhard 1993; Gerhard et al. 1998).

Quasi-separable distribution functions can be written as a product of two functions, one incorporating orbital energy dependence, the other incorporating orbital angular momentum dependence:

$$f(E, L) = g(E)j(h) \quad (3.1)$$

$$h = \frac{L}{L_0 + L_c(E)} \quad (3.2)$$

where $L_c(E)$ is the angular momentum of a circular orbit with energy E , and h is a parametrisation of angular momentum, L , known as the orbital circularity. Since all the angular momentum dependence is now contained within $j(h)$, for a fixed density profile, it is solely responsible for the velocity anisotropy profile, and hence is called the circularity function. This circularity function can be specified to have any functional dependence on h (subject to $j(h) \geq 0 \forall h$), in combination with the constant L_0 (subject to $L_0 \geq 0$) to allow a great variety of velocity anisotropy profiles. A function $j(h)$ decreasing with h gives radially biased velocity profiles, $j(h)$ increasing with h gives tangentially biased velocity profiles, and $j = \text{const}$ gives an isotropic velocity profile. The constant L_0 can be thought of as an anisotropy radius (the radius at which the velocity profile becomes anisotropic) times a characteristic speed, and represents a centripetal barrier around the core. More details on j and L_0 will be discussed further

after some definitions and derivation of required parameters.

3.3 Model Families

With a fixed tracer density profile selected, specifying the desired functional form of $j(h)$ then determines the velocity anisotropy profile. In general, the number density profile is related to the distribution function through $\nu = \int_{all\ E,L} g(E)j(h)dEdL$. Specific to quasi-separable distribution functions, $f(E, L) = g(E)j(h)$, and thus with the circularity function, $j(h)$, and the density profile, ν , also specified, the only unknown is $g(E)$ which must be solved for. All host galaxies are assumed to be isotropic, with their distribution functions and density profiles discussed below after some notation definitions and necessary equations.

$$\tilde{\Phi} = \frac{\Phi - \Phi_0}{\sigma_0^2} \quad (3.3)$$

$$\tilde{E} = \frac{E - \Phi_0}{\sigma_0^2} \quad (3.4)$$

$$\tilde{\rho} = \frac{\rho}{\rho_0} \quad (3.5)$$

$$\tilde{r} = \frac{r}{r_0} \quad (3.6)$$

$$\tilde{\nu} = \nu r_0^3 \quad (3.7)$$

$$\tilde{f} = \frac{f}{f_0} = f r_0^3 \sigma_0^3 \quad (3.8)$$

$$t_0 = \frac{r_0}{\sigma_0} = 1 \text{ Myr} \quad (3.9)$$

$$M_0 = \frac{r_0 \sigma_0^2}{G} \quad (3.10)$$

where σ_0 is some characteristic speed, such as for example the core velocity dispersion. With the characteristic time set as $t_0 = 1\text{Myr}$, only one of the pair of the characteristic radius, r_0 , and the characteristic speed, σ_0 , are free. Specifying either one will automatically assign a value to the other.

Before $g(E)$ can be solved for, the host galaxy potential is needed, as the tracers are essentially a massless population orbiting in the gravitational potential of a host galaxy. The host galaxy gravitational potential, Φ , and density profile, ρ , are related through Poisson's equation:

$$\frac{1}{r^2} \frac{d}{dr} \left(r^2 \frac{d\Phi}{dr} \right) = 4\pi G \rho \quad (3.11)$$

which may be solved to give the potential as a function of galactocentric radius, $\Phi(r)$, and therefore the density is known as a function of both galactocentric radius and gravitational potential. However, the models used for the host galaxy are expressed in dimensionless notation as given in equations (3.3 — 3.10), and so Poisson's equation must be solved in the same notation. This is achieved through substituting the definition

$$P_0 \equiv \frac{4\pi G \rho_0 r_0^2}{\sigma_0^2} \quad (3.12)$$

into Poisson's equation, to give the dimensionless version:

$$\frac{1}{\tilde{r}^2} \frac{d}{d\tilde{r}} \left(\tilde{r}^2 \frac{d\tilde{\Phi}}{d\tilde{r}} \right) = P_0 \tilde{\rho} \quad (3.13)$$

The constant P_0 simply controls the ratio between ρ_0 , r_0 , and σ_0 , and is fixed at 9 throughout (e.g. King 1966). This is because $P_0 = 9$ is often chosen by many other authors in the literature (as this choice makes r_0 close to the observed core radius of many systems, e.g. King 1966). Consequently setting $P_0 = 9$ here facilitates easier comparison of results.

When Poisson's equation is solved using the host galaxy densities given below, the first derivative of the gravitational potential, $d\tilde{\Phi}/d\tilde{r}$, is also obtained. There are two possible starting points for deriving the properties of the host galaxies. Either start with the host galaxy density profile which can be immediately plugged into Poisson's equation, then the resulting output used to solve an Abel integral equation for the isotropic distribution function of the host galaxy. Alternatively, it is possible to start with the isotropic distribution function of the host galaxy, and integrate it to get the

host galaxy density, which may then be used with Poisson's equation to obtain the host galaxy gravitational potential and its derivatives. These may then be used to calculate a range of other useful quantities that will be useful in solving for $\tilde{g}(\tilde{E})$.

The convention used throughout the rest of this thesis is that for non-singular models, ρ_0 and Φ_0 in equations (3.3) and (3.5) are the values of ρ and Φ at $\tilde{r} = 0$. For singular models, ρ_0 and Φ_0 are the values of ρ and Φ at $\tilde{r} = \tilde{r}_{ref}$, where $\tilde{r}_{ref} \equiv \tilde{r}_{init}/10$. Since the range of radii Poisson's equation is numerically solved over is $\tilde{r}_{init} \leq \tilde{r} \leq \tilde{r}_f$, this means that $\tilde{\rho} \leq 1$ and $\tilde{\Phi} \geq 0$ over the entire range of $\tilde{r}_{init} \leq \tilde{r} \leq \tilde{r}_f$ for both singular and non-singular models.

3.3.1 Polytropes

Polytropes are a family of models characterised by density profiles that are constant in the core, and turn over roughly to power-laws asymptotically. The polytropic equation of state relates pressure, P , and density, ρ , through the polytropic index, n , as $P \propto \rho^{(n+1)/n}$. The isotropic distribution function and density as a function of gravitational potential for polytropes are given by (e.g. Binney & Tremaine 2008):

$$\tilde{f} = \begin{cases} n(n-1) [-2\pi(n+1)] \frac{\Gamma(3/2-n)}{\Gamma(2-n)} \left(1 - \frac{\tilde{E}}{n+1}\right)^{n-3/2} & \text{for } n < -1 \\ n(n-1) [2\pi(n+1)] \frac{\Gamma(n-1)}{\Gamma(n-1/2)} \left(1 - \frac{\tilde{E}}{n+1}\right)^{n-3/2} & \text{for } n > -1 \end{cases}$$

$$\tilde{\rho} = \left(1 - \frac{\tilde{\Phi}}{n+1}\right)^n$$

Under the substitution (the reasons for which will become clear later on)

$$n = \frac{k}{k-2} \tag{3.14}$$

$$k = \frac{2n}{n-1} \tag{3.15}$$

the pressure and density are related by $P \propto \rho^{2(k-1)/k}$, and for a physical model both the pressure and density must decrease outwards, meaning $2(k-1)/k > 0$ and thus

$k > 1$ or $k < 0$. Writing the density and distribution function in terms of k gives

$$\tilde{\rho} = \left[1 - \frac{k-2}{2(k-1)} \tilde{\Phi} \right]^{\frac{k}{k-2}} \quad (3.16)$$

$$\tilde{f} = \begin{cases} \frac{2k}{(2-k)^2} \left[\frac{2-k}{4\pi(k-1)} \right]^{3/2} \frac{\Gamma(\frac{k}{2-k}+3/2)}{\Gamma(\frac{k}{2-k}+2)} \left(1 + \frac{2-k}{2(k-1)} \tilde{E} \right)^{\frac{k-6}{2(2-k)}} & \text{for } 1 < k < 2 \\ \frac{2k}{(k-2)^2} \left[\frac{k-2}{4\pi(k-1)} \right]^{3/2} \frac{\Gamma(\frac{k}{k-2}-1)}{\Gamma(\frac{k}{k-2}-1/2)} \left(1 - \frac{k-2}{2(k-1)} \tilde{E} \right)^{\frac{k-6}{2(2-k)}} & \text{for } k > 2 \end{cases} \quad (3.17)$$

As $k \rightarrow 2$, both forms of equation (3.17) converge to the same limiting form, and equation (3.16) also converges to a limiting form, given by:

$$\tilde{f} = (2\pi)^{-3/2} \exp(-\tilde{E}) \quad (3.18)$$

$$\tilde{\rho} = \exp(-\tilde{\Phi}) \quad (3.19)$$

namely, the isotropic distribution function and density of an isothermal sphere.

Making the substitution $\psi = 1 - \frac{k-2}{2(k-1)} \tilde{\Phi}$ and assuming a solution of the form $\psi = a\tilde{r}^b$ where a and b are constants to be determined, it is possible to combine $\tilde{\rho} = \psi^{k/(k-2)}$ with Poisson's equation to obtain

$$\tilde{\Phi} = \frac{2(k-1)}{k-2} \left[1 - \left(\frac{P_0}{2(k-1)(3-k)} \right)^{\frac{2-k}{2}} \tilde{r}^{2-k} \right] \quad (3.20)$$

as the asymptotic solution for $\tilde{\Phi}(\tilde{r})$ as $\tilde{r} \rightarrow \infty$ for $k \neq 2$. Then taking the limit of equation (3.20) as $k \rightarrow 2$ gives

$$\tilde{\Phi} = \ln \left(\frac{P_0 \tilde{r}^2}{2} \right) \quad (3.21)$$

as the asymptotic solution of $\tilde{\Phi}(\tilde{r})$ for $k = 2$. Equations (3.20) and (3.21) can be used to obtain $\tilde{\Phi}$ whenever it is required at galactocentric radii outside of the maximum radius Poisson's equation is numerically solved over, \tilde{r}_f .

By inserting these asymptotic approximations back into (3.16) for $k \neq 2$ or (3.19) for $k = 2$, it is plain that k is the asymptotic power dependence of $\tilde{\rho}(\tilde{r})$, i.e. $\tilde{\rho}(\tilde{r}) \propto \tilde{r}^{-k}$

as $\tilde{r} \rightarrow \infty$, and thus we must have that $k > 0$. From the polytropic equation of state, $k > 1$ or $k < 0$ are required, and thus including this result means that only $k > 1$ polytropes are allowed.

By performing a Taylor expansion of $\tilde{\rho}(\tilde{\Phi})$ about $\tilde{\Phi} = 0$ using either equation (3.16) for $k \neq 2$ or equation (3.19) for $k = 2$, combining the result with Poisson's equation and replacing all $\tilde{\Phi}$ with an assumed power series, $\tilde{\Phi} = \alpha\tilde{r} + \beta\tilde{r}^2 + \gamma\tilde{r}^3 + \dots$, then comparing coefficients and powers, it is possible to derive an analytic asymptote for $\tilde{\Phi}(\tilde{r})$ as $\tilde{r} \rightarrow 0$:

$$\tilde{\Phi} \simeq \frac{P_0}{3!}\tilde{r}^2 - \frac{kP_0^2}{5!2(k-1)}\tilde{r}^4 + \frac{k(3k+10)P_0^3}{7!12(k-1)^2}\tilde{r}^6 - \frac{k(9k^2+86k+280)P_0^4}{9!72(k-1)^3}\tilde{r}^8 + \dots \quad (3.22)$$

Equation (3.22) can be used to obtain $\tilde{\Phi}$ whenever it is required at galactocentric radii inside of the minimum radius Poisson's equation is solved over, \tilde{r}_{init} .

The case of $k = 2.5$ corresponds to $n = 5$, giving the well known Plummer (1911) sphere which has a fully analytical solution:

$$\tilde{\Phi} = 6 \left[1 - \left(1 + \frac{P_0}{18}\tilde{r}^2 \right)^{-1/2} \right] \quad (3.23)$$

$$\tilde{\rho} = \left(1 + \frac{P_0}{18}\tilde{r}^2 \right)^{-5/2} \quad (3.24)$$

Asymptotically, the Plummer sphere has $\tilde{\rho} \propto \tilde{r}^{-5}$ as $\tilde{r} \rightarrow \infty$, in contradiction to what k is understood to represent for polytropes. The reason for this is that at large \tilde{r} , the true solution actually oscillates around the asymptotic approximation, with both the amplitude and wave-number of the oscillations increasing sharply as $k \rightarrow 2.5$ (cf. Figure 3.1). At $k = 2.5$, a critical value is reached and the wave-number becomes infinite such that one complete oscillation would occur at $\tilde{r} = \infty$, and the trough occurs at one quarter of this value, which is obviously still infinite. So essentially, all that can be seen of the Plummer solution is the dive down to a trough at $\tilde{r} = \infty$, which is quite separate from the asymptotic approximation (e.g. Chandrasekhar 1942). This can be seen in Figure 3.1, where the numerical data and asymptote match very well for the black $k = 2.3$ lines, as opposed to the great disparity in the red $k = 2.5$ lines.

Polytropes with $k > 2.5$ have density gradients sufficiently steep so as to reach zero at a finite radius, but are not of much use as King (1966) models are used for constant density and finite zero density radii galaxies, such that $1 < k \leq 2.5$ for polytropes.

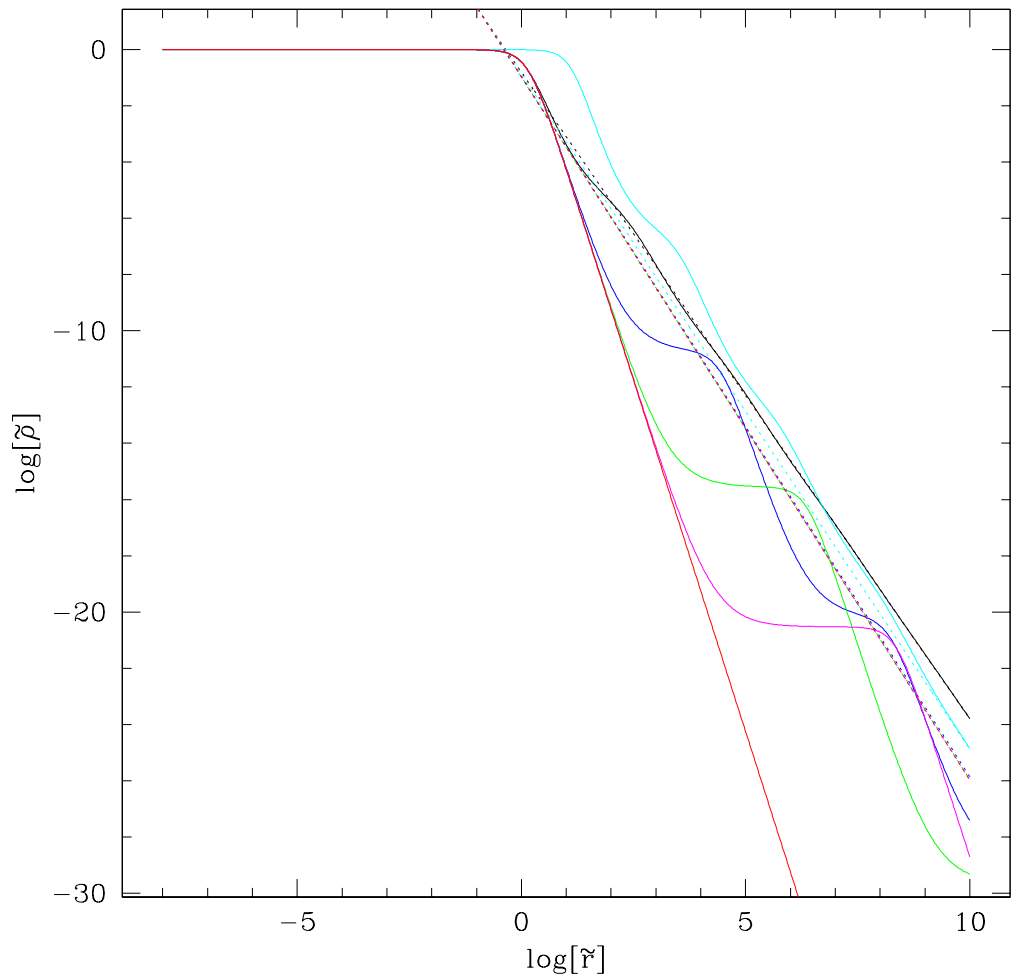


Figure 3.1: Density vs galactocentric radius for polytropes with $k \in [2.3, 2.5]$. Black: $k = 2.3$, cyan: $k = 2.4$, blue: $k = 2.49$, green: $k = 2.499$, magenta: $k = 2.4999$, red: $k = 2.5$. Solid lines are numerical data, dotted lines are asymptotes given by equation (3.20).

3.3.2 King Models

King (1966) models are often referred to as ‘lowered isothermal spheres’. This is because their velocity distribution is similar to that of an isothermal sphere, but with a finite escape velocity at all radii. Consequently their total mass is finite, and in fact King models are finite in extent as well as in mass. This outermost radius is often referred to as the tidal radius. However, in this context the host galaxy is being described by a King model, and a tidally limited host galaxy is not being considered. Therefore, the outermost radius of a King model will be referred to as a zero density radius. Due to the finite extent of these models, they provide good fits to structures with clear edges, such as globular clusters. Since King models were originally conceived to describe the spatial and velocity distributions of clusters, there is no physical reason to use King models to describe a host galaxy, however they are useful functional forms to apply. Density as a function of gravitational potential and the isotropic distribution function for King models are given by:

$$\tilde{\rho} = \frac{\exp(W_0 - \tilde{\Phi}) \operatorname{erf}(\sqrt{W_0 - \tilde{\Phi}}) - \sqrt{\frac{4(W_0 - \tilde{\Phi})}{\pi}} \left(1 + \frac{2(W_0 - \tilde{\Phi})}{3}\right)}{\exp(W_0) \operatorname{erf}(\sqrt{W_0}) - \sqrt{\frac{4W_0}{\pi}} \left(1 + \frac{2W_0}{3}\right)} \quad (3.25)$$

$$\tilde{f} = \frac{(2\pi)^{-3/2}}{\exp(W_0) \operatorname{erf}(\sqrt{W_0}) - \sqrt{\frac{4W_0}{\pi}} \left(1 + \frac{2W_0}{3}\right)} \left[\exp(W_0 - \tilde{E}) - 1\right] \quad (3.26)$$

where W_0 is the value of the potential at the edge of the system, commonly referred to as the ‘central potential’, so named for the conventional definition of potential used in King models, where the potential is maximum in the centre and decreases outwards ($W = W_0 - \tilde{\Phi}$). Factoring out $\exp(W_0)$ and ignoring constants, the functional dependence of equation (3.26) is $\left[\exp(-\tilde{E}) - \exp(-W_0)\right]$, and so the limit as $W_0 \rightarrow \infty$ is simply $\exp(-\tilde{E})$. This is exactly the same as the functional dependence of equation (3.18), the isotropic distribution function of the isothermal sphere. Thus, a King model with $W_0 \rightarrow \infty$ is a $k = 2$ polytrope.

Analogously with polytropes, by performing a Taylor expansion of $\tilde{\rho}(\tilde{\Phi})$ about

$\tilde{\Phi} = 0$ using equation (3.25), combining the result with Poisson's equation and replacing all $\tilde{\Phi}$ with an assumed power series, $\tilde{\Phi} = \alpha\tilde{r} + \beta\tilde{r}^2 + \gamma\tilde{r}^3 + \dots$, then comparing coefficients and powers, it is possible to derive an analytic asymptote for $\tilde{\Phi}(\tilde{r})$ as $\tilde{r} \rightarrow 0$:

$$\tilde{\Phi} = \frac{P_0}{3!}\tilde{r}^2 + \frac{P_0 A}{5!}\tilde{r}^4 + \left[A^2 + \frac{10B}{3}\right] \frac{P_0^3}{7!}\tilde{r}^6 + \left[A^3 + \frac{52AB}{3} + \frac{70C}{3}\right] \frac{P_0^4}{9!}\tilde{r}^8 + \dots \quad (3.27)$$

$$\begin{aligned} A &= \frac{\exp(W_0) \operatorname{erf}(\sqrt{W_0}) - 2\sqrt{\frac{W_0}{\pi}}}{\exp(W_0) \operatorname{erf}(\sqrt{W_0}) - \sqrt{\frac{4W_0}{\pi}} \left(1 + \frac{2W_0}{3}\right)} \\ B &= \frac{\exp(W_0) \operatorname{erf}(\sqrt{W_0})}{2 \left[\exp(W_0) \operatorname{erf}(\sqrt{W_0}) - \sqrt{\frac{4W_0}{\pi}} \left(1 + \frac{2W_0}{3}\right) \right]} \\ C &= -\frac{\exp(W_0) \operatorname{erf}(\sqrt{W_0}) + \frac{1}{\sqrt{W_0\pi}} - \frac{1}{\sqrt{4\pi W_0^3}}}{24 \left[\exp(W_0) \operatorname{erf}(\sqrt{W_0}) - \sqrt{\frac{4W_0}{\pi}} \left(1 + \frac{2W_0}{3}\right) \right]} \end{aligned}$$

These small \tilde{r} expansions for $\tilde{\Phi}$ are useful for when $\tilde{\Phi}$ is required very precisely at small galactocentric radii, where variations in numerical data may become blurred in numerical noise.

3.3.3 Dehnen Models

The Dehnen (1993) family of models are essentially two power-law models, with a density gradient determined by the parameter γ at inner radii, that turns over to a $\tilde{\rho} \propto \tilde{r}^{-4}$ as $\tilde{r} \rightarrow \infty$, for all γ . Due to the steep density gradient at large radii, all Dehnen family models have finite total masses. Most properties of Dehnen models are analytic, except for the distribution function for certain values of γ . The density, gravitational potential, and isotropic distribution function for Dehnen models are given in the notation of equations (3.3 to 3.10) below:

$$\tilde{\rho} = 2^{4-\gamma} \tilde{r}^{-\gamma} (1 + \tilde{r})^{\gamma-4} \quad (3.28)$$

$$\tilde{\Phi} = \begin{cases} 4P_0 \ln \left(\frac{\tilde{r}}{1+\tilde{r}} \frac{1+\tilde{r}_{ref}}{\tilde{r}_{ref}} \right) & \text{for } \gamma = 2 \\ \frac{2^{4-\gamma} P_0}{(3-\gamma)(2-\gamma)} \left[\left(\frac{\tilde{r}}{1+\tilde{r}} \right)^{2-\gamma} - \left(\frac{\tilde{r}_{ref}}{1+\tilde{r}_{ref}} \right)^{2-\gamma} \right] & \text{for } \gamma \neq 2 \end{cases} \quad (3.29)$$

$$\tilde{f} = \begin{cases} \left(\frac{3-\gamma}{P_0}\right)^{3/2} \frac{2^{(\gamma-5)/2}}{\pi^2} \int_{u_E}^1 \frac{(4-\gamma)u^4 + 2u^3(\gamma-3) + 2u(1-\gamma) + \gamma}{u^3 \sqrt{\frac{u^{2-\gamma} - u_E^{2-\gamma}}{2-\gamma}}} du & \text{for } \gamma \neq 2 \\ \frac{1}{(2P_0^3)^{1/2}\pi^2} \int_{u_E}^1 \frac{u^4 - u^3 - u + 1}{u^3 \sqrt{\ln(\frac{u}{u_E})}} du & \text{for } \gamma = 2 \end{cases} \quad (3.30)$$

$$u_E = \begin{cases} \left[\frac{(3-\gamma)(2-\gamma)\tilde{E}}{2^{4-\gamma}P_0} + \left(\frac{\tilde{r}_{ref}}{1+\tilde{r}_{ref}}\right)^{2-\gamma} \right]^{\frac{1}{2-\gamma}} & \text{for } \gamma \neq 2 \\ \frac{\tilde{r}_{ref}}{1+\tilde{r}_{ref}} \exp\left(\frac{\tilde{E}}{4P_0}\right) & \text{for } \gamma = 2 \end{cases}$$

where \tilde{r}_{ref} is the radius at which $\tilde{\Phi} = 0$, defined by $\tilde{r}_{ref} \equiv \tilde{r}_{init}/10$, so that $\tilde{\Phi} > 0$ for all radii $\tilde{\Phi}$ is calculated over. By inspection of equation (3.28), it is clear that γ is the logarithmic slope of the inner cusp, and thus we must have $\gamma \geq 0$. Furthermore, to have a finite central mass, we must have $\gamma < 3$, and consequently $0 \leq \gamma < 3$ for these models. The Dehnen family of models also includes the Hernquist (1990) and Jaffe (1983) models as special cases, given by $\gamma = 1$ and $\gamma = 2$ respectively.

3.3.4 Power-law Models

Power-laws are very simple and completely analytic models, and thus are very popular. They are characterised by a constant logarithmic density slope at all galactocentric radii. A $k = 2$ power-law gives the singular isothermal sphere, or logarithmic potential as it is also known, with constant rotation speed at all radii. Furthermore, $k = 2$ denotes the transition from finite to infinite total mass. The density, gravitational potential, and isotropic distribution function for power-law models are given by:

$$\tilde{\rho} = \tilde{r}^{-k} \quad (3.31)$$

$$\tilde{\Phi} = \begin{cases} P_0 \ln\left(\frac{\tilde{r}}{\tilde{r}_{ref}}\right) & \text{for } k = 2 \\ \frac{P_0}{(3-k)(2-k)} (\tilde{r}^{2-k} - \tilde{r}_{ref}^{2-k}) & \text{for } k \neq 2 \end{cases} \quad (3.32)$$

$$\tilde{f} = \begin{cases} \sqrt{\frac{k^2}{2(2-k)}} \left(\frac{3-k}{\pi P_0}\right)^{3/2} \frac{\Gamma(\frac{k}{2-k}+3/2)}{\Gamma(\frac{k}{2-k}+2)} \left[\frac{(3-k)(2-k)\tilde{E}}{P_0} + \tilde{r}_{ref}^{2-k} \right]^{\frac{k-6}{2(2-k)}} & \text{for } k < 2 \\ \tilde{r}_{ref}^{-2} (P_0 \pi)^{-3/2} \exp\left(\frac{-2\tilde{E}}{P_0}\right) & \text{for } k = 2 \\ \sqrt{\frac{k^2}{2(k-2)}} \left(\frac{3-k}{\pi P_0}\right)^{3/2} \frac{\Gamma(\frac{k}{k-2}-1)}{\Gamma(\frac{k}{k-2}-1/2)} \left[\tilde{r}_{ref}^{2-k} - \frac{(3-k)(k-2)\tilde{E}}{P_0} \right]^{\frac{k-6}{2(2-k)}} & \text{for } k > 2 \end{cases} \quad (3.33)$$

where similarly to Dehnen models, the notation \tilde{r}_{ref} has been defined as the radius at which $\tilde{\Phi} = 0$, selected such that $\tilde{\Phi} > 0$ for all $\tilde{r} > \tilde{r}_{init}$, and k is the logarithmic slope of density throughout. Therefore, k must be bounded by $0 < k < 3$ to have a negative density gradient and a finite central mass. However, the functional form of the $k < 2$, $k = 2$, and $k > 2$ distribution functions are the same as those of polytropes (hence the existence of singular power-law solutions for polytropes), and thus $k > 1$ is required for power-laws the same as for polytropes, giving $1 < k < 3$ for power-laws.

3.4 Galaxy Parameters

Equation (3.13) is solved numerically over a range of radii (from \tilde{r}_{init} to \tilde{r}_f) to give dimensionless values for the potential and first derivative of potential with respect to galactocentric radius. When these are combined with the already known density, this also gives the second derivative of potential with respect to galactocentric radius.

Dimensionless values for angular momentum of a circular orbit with radius \tilde{r} , \tilde{L}_c , mass interior to radius \tilde{r} , \tilde{M} , circular speed at radius \tilde{r} , \tilde{v}_c , radial component of acceleration at radius \tilde{r} , \tilde{a}_r , circular angular frequency at radius \tilde{r} , $\tilde{\Omega}_c$, and finally

energy of a circular orbit with radius \tilde{r} , \tilde{E}_c , are then given by:

$$\tilde{L}_c^2 = \tilde{r}^3 \frac{d\tilde{\Phi}}{d\tilde{r}} \quad (3.34)$$

$$\tilde{M} = \tilde{r}^2 \frac{d\tilde{\Phi}}{d\tilde{r}} \quad (3.35)$$

$$\tilde{v}_c^2 = \tilde{r} \frac{d\tilde{\Phi}}{d\tilde{r}} \quad (3.36)$$

$$\tilde{a}_r = \frac{d\tilde{\Phi}}{d\tilde{r}} \quad (3.37)$$

$$\tilde{\Omega}^2 = \frac{1}{\tilde{r}} \frac{d\tilde{\Phi}}{d\tilde{r}} \quad (3.38)$$

$$\tilde{E}_c = \tilde{\Phi} + \frac{\tilde{r}}{2} \frac{d\tilde{\Phi}}{d\tilde{r}} \quad (3.39)$$

When each of these parameters is calculated at the same radii, any of these parameters can be thought of as a function of any of the others, e.g. $\tilde{L}_c(\tilde{E}_c)$.

The velocity dispersion of the host galaxy can be calculated by solving the spherical Jeans equation:

$$\frac{d\tilde{\rho}\tilde{\sigma}_r^2}{d\tilde{r}} + 2\frac{\beta\tilde{\rho}\tilde{\sigma}_r^2}{\tilde{r}} = -\tilde{\rho}\frac{d\tilde{\Phi}}{d\tilde{r}} \quad (3.40)$$

where β is the anisotropy parameter defined in terms of the radial and tangential velocity dispersions, given by:

$$\beta(\tilde{r}) = 1 - \frac{\tilde{\sigma}_t^2}{2\tilde{\sigma}_r^2} \quad (3.41)$$

In the case that the velocity distribution is isotropic, $\tilde{\sigma}_r^2 = \tilde{\sigma}_\theta^2 = \tilde{\sigma}_\phi^2 = \frac{1}{2}\tilde{\sigma}_t^2 \equiv \tilde{\sigma}^2$, and thus $\beta = 0$. Then only the spherical isotropic Jeans equation need be solved:

$$\frac{d(\tilde{\rho}\tilde{\sigma}^2)}{d\tilde{r}} = -\tilde{\rho}\frac{d\tilde{\Phi}}{d\tilde{r}} \quad (3.42)$$

Then given that only models satisfying $\lim_{\tilde{r} \rightarrow \infty} (\tilde{\rho}\tilde{\sigma}^2) = 0$ are allowed, the isotropic velocity dispersion is given by:

$$\tilde{\sigma}^2 = \frac{1}{\tilde{\rho}} \int_{\tilde{r}}^{\infty} \tilde{\rho} \frac{d\tilde{\Phi}}{d\tilde{r}} d\tilde{r} \quad (3.43)$$

For models where $\tilde{\rho}$ and $\frac{d\tilde{\Phi}}{d\tilde{r}}$ as functions of \tilde{r} must be obtained numerically, the integration of equation (3.43) would normally be impossible beyond the outermost radius, \tilde{r}_f . However, out of the model types used this only potentially poses any difficulty for polytropes, as a King model with zero density radius $\rightarrow \infty$ is just an isothermal sphere. Fortunately, polytropes have power-law solutions for $\tilde{\Phi}$ and $\tilde{\rho}$ as functions of \tilde{r} as $\tilde{r} \rightarrow \infty$ (see section 3.3.1), allowing calculation of $\tilde{\rho}$ and $\frac{d\tilde{\Phi}}{d\tilde{r}}$ up to arbitrarily high radii for the integration in equation (3.43). The host galaxy velocity dispersion will be useful for the calculation of dynamical friction and the Coulomb logarithm in Chapter 5.

In order to proceed to the calculation of the functions that will ultimately be used in the Monte Carlo sampling, some more definitions must be made. The dimensionless tangential velocity \tilde{v}_t is defined such that:

$$\tilde{v}_t = \sqrt{\tilde{v}_\phi^2 + \tilde{v}_\theta^2} = \frac{\tilde{L}}{\tilde{r}} \quad (3.44)$$

$$\tilde{v}_\phi = \tilde{v}_t \sin \gamma_{inc} \quad (3.45)$$

$$\tilde{v}_\theta = \tilde{v}_t \cos \gamma_{inc} \quad (3.46)$$

$$\tilde{E} = \tilde{\Phi}(\tilde{r}) + \frac{1}{2} \left(\tilde{v}_r^2 + \frac{\tilde{L}^2}{\tilde{r}^2} \right) \quad (3.47)$$

where \tilde{v}_r is the component of velocity in the radial direction, γ_{inc} and \tilde{v}_t are the direction and magnitude of the resultant vector of \tilde{v}_ϕ and \tilde{v}_θ on a plane perpendicular to \tilde{v}_r , and \tilde{E} is the orbital energy of a particle with orbital angular momentum \tilde{L} and radial speed \tilde{v}_r at radius \tilde{r} . Note that with this definition, \tilde{v}_t is strictly positive. When the orbital energy, angular momentum, and instantaneous galactocentric radius of a particle are known, its radial speed can be calculated via

$$\tilde{v}_r = \pm \sqrt{2 \left[\tilde{E} - \tilde{\Phi}(\tilde{r}) \right] - \frac{\tilde{L}^2}{\tilde{r}^2}} \quad (3.48)$$

where a positive radial speed denotes movement away from the galactic centre, and a negative radial speed denotes movement towards the galactic centre.

3.4.1 Integral Equations

By definition, the tracer population number density, ν , is related to the tracer population distribution function, $f(E, L)$, via

$$\begin{aligned}\nu &= \int_{\text{all } v} f(E, L) d^3v \\ \tilde{\nu} &= \int_{\text{all } \tilde{v}} \tilde{f}(\tilde{E}, \tilde{L}) d\tilde{v}_r d\tilde{v}_\theta d\tilde{v}_\phi\end{aligned}$$

where on the second line the equation has been made dimensionless analogously as in section 3.3, i.e. $\tilde{\nu} = r_0^3 \nu$. Changing coordinates from $d\tilde{v}_\theta, d\tilde{v}_\phi$ to $d\tilde{v}_t, d\gamma_{inc}$, the Jacobian determinant is given by:

$$d\tilde{v}_\theta d\tilde{v}_\phi = \begin{vmatrix} \frac{\partial \tilde{v}_\theta}{\partial \tilde{v}_t} & \frac{\partial \tilde{v}_\theta}{\partial \gamma_{inc}} \\ \frac{\partial \tilde{v}_\phi}{\partial \tilde{v}_t} & \frac{\partial \tilde{v}_\phi}{\partial \gamma_{inc}} \end{vmatrix} d\tilde{v}_t d\gamma_{inc}$$

Using equations (3.45) and (3.46), this gives $d\tilde{v}_\theta d\tilde{v}_\phi = \tilde{v}_t d\tilde{v}_t d\gamma_{inc}$. Thus

$$\tilde{\nu} = \int_{-\tilde{v}_{max}}^{\tilde{v}_{max}} d\tilde{v}_r \int_0^{\sqrt{\tilde{v}_{max}^2 - \tilde{v}_r^2}} \tilde{v}_t d\tilde{v}_t \int_0^{2\pi} \tilde{f}(\tilde{E}, \tilde{L}) d\gamma_{inc}$$

where \tilde{v}_{max} is the maximum speed a particle can have and remain bound to the system, i.e. such that the total orbital energy is sufficient to reach the edge of the system, $\tilde{E} = \tilde{\Phi}(\tilde{r}) + \frac{1}{2}\tilde{v}_{max}^2 = \tilde{\Phi}(\tilde{r}_f)$, and thus this defines the maximum energy attainable for any bound particle, $\tilde{E}_{max} = \tilde{\Phi}(\tilde{r}_f)$. With the assumption of spherical symmetry (i.e. $\tilde{f}(\tilde{E}, \tilde{L})$ is independent of γ_{inc} and is an even function of \tilde{v}_r), the integral reduces to

$$\tilde{\nu} = 4\pi \int_0^{\tilde{v}_{max}} d\tilde{v}_r \int_0^{\sqrt{\tilde{v}_{max}^2 - \tilde{v}_r^2}} \tilde{f}(\tilde{E}, \tilde{L}) \tilde{v}_t d\tilde{v}_t$$

The lowest speed a particle can have is to be stationary (i.e. a radial orbit at pericentre or apocentre), thus the lowest attainable energy is $\tilde{E}_{min} = \tilde{\Phi}$, and as discussed above the highest energy attainable is \tilde{E}_{max} . For a particle with only orbital energy and instantaneous galactocentric position known, the maximum angular momentum possible is $L_{max} = \tilde{r}\tilde{v} = \tilde{r}\sqrt{2(\tilde{E} - \tilde{\Phi})}$, i.e. if the particle were on a circular orbit.

The lowest angular momentum possible is that of a radial orbit, $\tilde{L}_{min} = 0$. Again changing coordinates, this time from $d\tilde{v}_r d\tilde{v}_t$ to $d\tilde{E} d\tilde{L}$, with equations (3.44) and (3.48) the Jacobian determinant gives $d\tilde{v}_r d\tilde{v}_t = \frac{1}{\tilde{r}\tilde{v}_r} d\tilde{E} d\tilde{L}$. Then

$$\tilde{\nu} = 4\pi \int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \int_0^{\tilde{r} \left[2(\tilde{E} - \tilde{\Phi}) \right]^{1/2}} \tilde{f}(\tilde{E}, \tilde{L}) \frac{\tilde{v}_t}{\tilde{r}\tilde{v}_r} d\tilde{L}$$

Then with the definition in equation (3.2) (i.e., $h = \tilde{L}/[\tilde{L}_0 + \tilde{L}_c(\tilde{E})]$), we may introduce the equivalent maximum dimensionless angular momentum:

$$h_{max} = \frac{\left[2(\tilde{E} - \tilde{\Phi}) \right]^{1/2} \tilde{r}(\tilde{\Phi})}{\tilde{L}_0 + \tilde{L}_c(\tilde{E})} \quad (3.49)$$

Using equations (3.2), (3.44), (3.48), and (3.49), the integral may be written as

$$\tilde{\nu}(\tilde{\Phi}) = \frac{4\pi}{\tilde{r}(\tilde{\Phi})} \int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \int_0^{h_{max}} \frac{\tilde{f}(\tilde{E}, h) h \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^2 dh}{\tilde{r}^2 \left(\frac{\left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^2 h_{max}^2}{\tilde{r}^2} - \frac{\left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^2 h^2}{\tilde{r}^2} \right)^{1/2}}$$

Finally, with the definition of quasi-separable distribution functions in equation (3.1), this may be tidied up and written as (e.g. Gerhard 1991):

$$\tilde{\nu}(\tilde{\Phi}) = \frac{4\pi}{\tilde{r}(\tilde{\Phi})} \int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right] \int_0^{h_{max}} \frac{h j(h)}{\sqrt{h_{max}^2 - h^2}} dh \quad (3.50)$$

In this notation, the radial and tangential velocities are given by:

$$\tilde{v}_r^2 = \left(\frac{\tilde{L}_0 + \tilde{L}_c(\tilde{E})}{\tilde{r}(\tilde{\Phi})} \right)^2 \left[h_{max}^2(\tilde{E}, \tilde{\Phi}) - h^2 \right] \quad (3.51)$$

$$\tilde{v}_t^2 = \left(\frac{\tilde{L}_0 + \tilde{L}_c(\tilde{E})}{\tilde{r}(\tilde{\Phi})} \right)^2 h^2 \quad (3.52)$$

Equation (3.50) now makes it clear how \tilde{L}_0 represents a centripetal barrier; for small orbital energies, $\tilde{L}_c(\tilde{E})$ is also small, and therefore $\tilde{L} < \tilde{L}_c(\tilde{E}) \ll \tilde{L}_0$, so that $h_{max} \sim 0$, such that low-energy orbits contribute very little to the density. Additionally, equation

(3.50) also makes it clear how $j(h)$ increasing with h makes tangentially biased velocity profiles: with $j(h)$ larger for values of h near to h_{max} , more density is contributed by orbits with angular momenta near to that required for circular orbits, whereas for a function $j(h)$ that initially has a larger value for low h and decreases, more density is contributed by orbits with small angular momenta. A more thorough analysis of the effects of the circularity function and \tilde{L}_0 on the velocity profile is given in section 4.2.

Note that $\tilde{\nu}$ in equation (3.50) is distinct from $\tilde{\rho}$ used with Poisson's equation (equation 3.13) to derive the potential, as $\tilde{g}(\tilde{E})j(h)$ can be thought of as describing the phase distribution of an effectively massless tracer population orbiting in the potential generated by some host galaxy. As such, $\tilde{\nu}$ is actually a number density rather than a mass density (even in the case of direct tracing — i.e. where the tracer spatial distribution exactly follows that of the host galaxy). The procedure is to choose the tracer density profile, $\tilde{\nu}$, and the circularity function $j(h)$, and solve for whatever $\tilde{g}(\tilde{E})$ these distributions imply (so long as the solution is physical, i.e. $\tilde{g}(\tilde{E}) \geq 0$).

At this point, equation (3.50) is a Volterra integral equation of the first kind with solution $\tilde{g}(\tilde{E})$ and kernel $[\tilde{L}_0 + \tilde{L}_c(\tilde{E})] \int_0^{h_{max}} \frac{hj(h)}{\sqrt{h_{max}^2 - h^2}} dh$. The solution $\tilde{g}(\tilde{E})$ is obtained by using numerical quadrature, which breaks the integral over $[\tilde{\Phi}, \tilde{E}_{max}]$ into \mathcal{N} tiny integrals over $[\tilde{\Phi}_i, \tilde{\Phi}_{i+1}]$, with $\tilde{\Phi}_1 = \tilde{\Phi}$ and $\tilde{\Phi}_{\mathcal{N}} = \tilde{E}_{max}$, and uses quadrature to sequentially solve each tiny integral for $\tilde{g}(\tilde{E} = \tilde{\Phi}_i)$ (see Appendix B). However, when the upper limit on equation (3.50) is formally infinite (i.e. for $k \leq 2$ power-laws, $k \leq 2$ polytropes, and $W_0 \rightarrow \infty$ King models), the density has to be assumed to go to zero at a finite potential, since the range over which $\tilde{g}(\tilde{E})$ is solved numerically has to be finite; $\tilde{\Phi} \in [\tilde{\Phi}(\tilde{r}_{init}), \tilde{\Phi}(\tilde{r}_f)]$. This means that $\tilde{g}(\tilde{E})$ is forced to be zero over the range $[\tilde{\Phi}(\tilde{r}_f), \infty]$. The result of this is that tracers orbiting in galaxy models with potentials that would normally increase to infinity cannot trace the galaxy out to infinite galactocentric radii, and are instead truncated at some finite potential (\tilde{E}_{max}), with the density going smoothly to zero there. This finite upper limit on the potential is taken to be the value of the potential at a radius \tilde{r}_f from the original unaltered model, so that r_f can be made large enough such that the properties of the truncated

models are identical to those of the original models over any sensible range of radii a tracer population could be modelled, and only differ at radii so large that it would be questionable to consider tracer objects at these radii to still be associated with the host galaxy anyway.

3.5 Tracer System Properties

As the tracer population is simply orbiting in the potential generated by the host galaxy, the potential and any derivatives as a function of radius are just the same as those of the host galaxy. The number of tracers interior to radius \tilde{r} for the tracer system (as opposed to mass for the galaxy) is given by:

$$N(\tilde{r}) = 4\pi \int_0^{\tilde{r}} \tilde{r}^2 \tilde{\nu}(\tilde{r}) d\tilde{r} \quad (3.53)$$

Radial and tangential velocity dispersions are obtained by taking the first moments of \tilde{v}_r^2 and \tilde{v}_t^2 , calculated by including the radial and tangential speeds as functions of $\tilde{E}, \tilde{\Phi}, h$ (equations 3.51 and 3.52) in the integrand of the distribution function (3.50):

$$\tilde{\sigma}_r^2(\tilde{\Phi}) = \frac{4\pi}{\tilde{\nu}(\tilde{\Phi})\tilde{r}^3(\tilde{\Phi})} \int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^3 \int_0^{h_{max}} h j(h) \sqrt{h_{max}^2 - h^2} dh \quad (3.54)$$

$$\tilde{\sigma}_t^2(\tilde{\Phi}) = \frac{4\pi}{\tilde{\nu}(\tilde{\Phi})\tilde{r}^3(\tilde{\Phi})} \int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^3 \int_0^{h_{max}} \frac{h^3 j(h) dh}{\sqrt{h_{max}^2 - h^2}} \quad (3.55)$$

3.5.1 The Circularity Function

Taking the definition of the anisotropy parameter (equation 3.41) and substituting for the tangential and radial velocity dispersions (equations 3.54 and 3.55), we can express

β as a function of $\tilde{g}(\tilde{E})$ and $j(h)$:

$$\beta(\tilde{\Phi}) = 1 - \frac{\int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^3 \int_0^{h_{max}} \frac{h^3 j(h) dh}{\sqrt{h_{max}^2 - h^2}}}{2 \int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^3 \int_0^{h_{max}} \sqrt{h_{max}^2 - h^2} j(h) dh} \quad (3.56)$$

Each model type has a different sequence of distribution functions (i.e. along k for power-laws and polytropes, along γ for Dehnen models, and along W_0 for King models), and thus even if the same $j(h)$ were used between model types, the different forms of $\tilde{g}(\tilde{E})$ would result in different anisotropy profiles (cf. equation 3.56). Thus the converse is also true — selecting a function $\beta(\tilde{r})$ and inverting to solve for $j(h)$ would result in different $j(h)$ for each model type, and need not even result in a physical solution (i.e. $j(h) \geq 0 \forall h$). Moreover, if $j(h)$ were an unknown to be solved for, then $\tilde{g}(\tilde{E})$ would also be an unknown as the above method for solving for $\tilde{g}(\tilde{E})$ will not work without a specified $j(h)$. Thus both $\tilde{g}(\tilde{E})$ and $j(h)$ would need to be solved for simultaneously. Obviously this would prove extremely difficult, and given the lack of guarantee for a physical solution, was consequently not attempted.

However, there is one anisotropy profile where an inversion is both possible and analytic; the case of constant anisotropy. For this to work, all that is required is for the numerator and denominator in equation (3.56) to have the same functional form, and thus no explicit dependence on the form of $\tilde{g}(\tilde{E})$. For all model types, constant anisotropy $\beta = \beta_0 < 1$ is achieved with a circularity function of the form:

$$j(h) = Ch^{-2\beta_0} \quad (3.57)$$

where C is an arbitrary constant (see Appendix C). Clearly $\beta_0 > 0$ gives a $j(h)$ decreasing with h , while $\beta_0 < 0$ gives a $j(h)$ increasing with h , in agreement with the expectation that a $j(h)$ decreasing with h gives radially biased velocity profiles, and a $j(h)$ increasing with h gives tangentially biased velocity profiles. Equation (3.57) is displayed with $\beta_0 = -1$ as the black curve in Figure 3.2.

Another natural choice of $j(h)$ to consider is:

$$j(h) = \exp(-ah) \quad (3.58)$$

since radially biased velocity profiles seem to be the way to achieve a GCMF invariant with galactocentric distance, and $j(h) \leq 1 \forall h$ is required later on for the sampling (see section 3.6), only $a > 0$ is considered for this circularity function. Equation (3.58) is displayed with $a = 2$ as the red curve in Figure 3.2.

A further natural choice for $j(h)$ is:

$$j(h) = \exp \left[-\frac{(h-a)^2}{2b^2} \right] \quad (3.59)$$

for $a = 0$ this would give a circularity function that decreases with h , while $0 < a < 1$ would give a circularity function that initially increases with h before peaking and decreasing again, and $a = 1$ would give a circularity function that increases with h . Thus, a Gaussian circularity function could be used to make either tangentially or radially biased velocity profiles, or presumably an anisotropy profile with some sort of peak. Moreover, b may be varied to change how quickly $j(h)$ increases/decreases with h , which presumably would affect how quickly the anisotropy profile becomes biased. Equation (3.59) is displayed with $a = 0.5, b = 0.2$ as the green curve in Figure 3.2.

In order to have a greater variety of circularity functions giving radially biased velocity profiles, the following equation for $j(h)$ is used:

$$j(h) = 1 - h^a [1 - \ln(h^a)] \quad (3.60)$$

for all $a > 0$, this function decreases from $j(0) = 1$ to $j(1) = 0$. Smaller values of a cause $j(h)$ to decrease more quickly initially, while larger a give a more gradual decrease. Equation (3.60) is displayed with $a = 2$ as the blue curve in Figure 3.2.

One more circularity function is defined, this time using a trigonometric function, to facilitate investigating the effects of an oscillating circularity function on the velocity profile.

$$j(h) = 0.5 [1 + h \sin(ah)] \quad (3.61)$$

for $a < 0$, this circularity function initially decreases with h , and an exact mirror image is obtained if $|a|$ is used instead. The amplitude of the oscillations of this circularity function increase with h , and for larger $|a|$, more oscillations are obtained over $0 \leq$

$h \leq 1$. Alternatively, this function could be used with smaller $|a|$ as a circularity function that either gently increases or decreases without oscillating. Equation (3.61) is displayed with $a = 4\pi$ as the magenta curve in Figure 3.2.

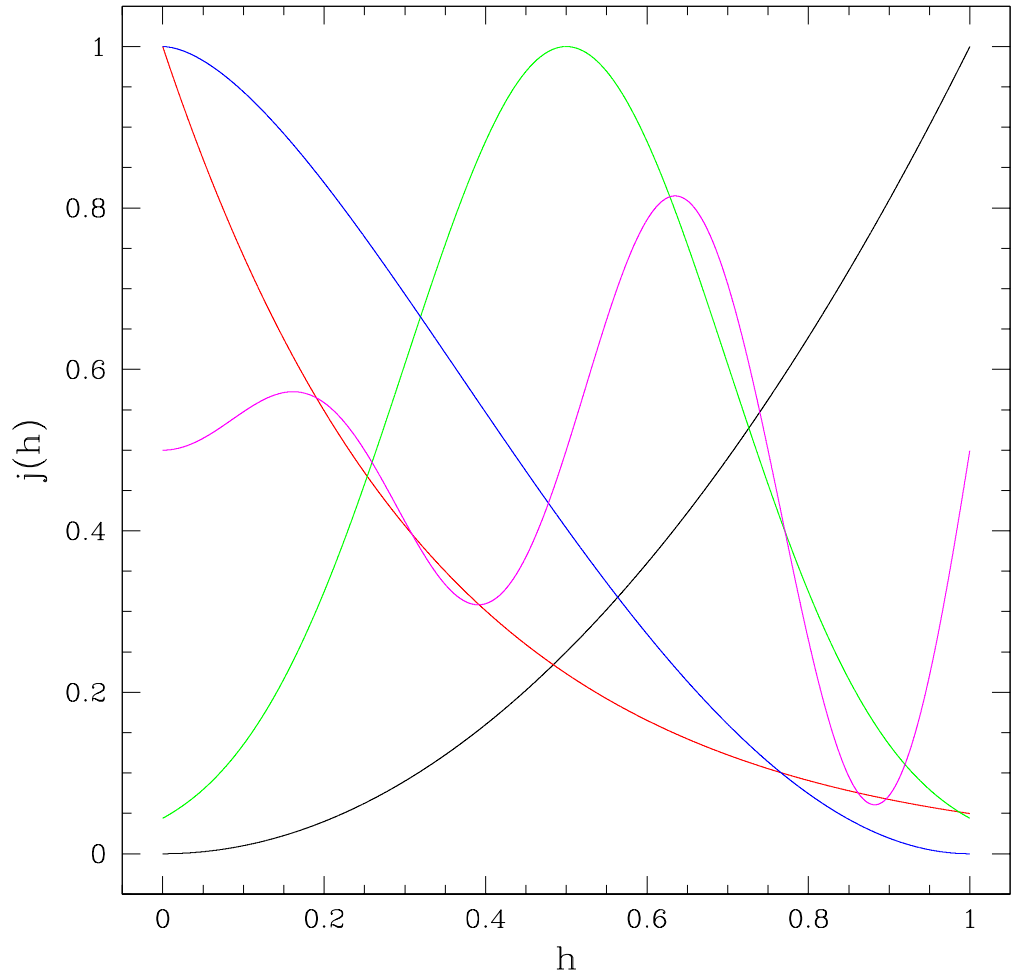


Figure 3.2: Various choices of circularity function against h . Black curve: equation (3.57) with $\beta_0 = -1$. Red curve: equation (3.58) with $a = 2$. Green curve: equation (3.59) with $a = 0.5, b = 0.2$. Blue curve: equation (3.60) with $a = 2$. Magenta curve: equation (3.61) with $a = 4\pi$.

3.6 Tracer Object Generation

Having obtained the tracer distribution function energy dependence, $\tilde{g}(\tilde{E})$, it is then possible to create a population of randomly generated objects with assigned properties in such a way that the ensemble properties match the function governing their distribution. This process is described for the generation of tracer masses, galactocentric positions, orbital energies, and angular momenta. Note that the random numbers used for these samplings are uniform between 0 and 1 exclusive (meaning that any number between 0 and 1 are equally likely to be drawn, but 0 and 1 themselves are excluded).

Gaining masses for randomly generated tracer objects is by far the simplest property to obtain, as it is completely independent of the other properties. Defining the dimensionless tracer particle mass to be $\tilde{M}_c = M_c/\bar{m}_0$, where \bar{m}_0 is the initial average stellar mass inside a tracer GC, then given an initial tracer mass function, dN/dM , to describe the number of tracer objects per unit mass between the lower and upper limits, $\tilde{M}_{c,min}$ and $\tilde{M}_{c,max}$, the cumulative distribution is given by:

$$n(< \tilde{M}_c) = \int_{\tilde{M}_{c,min}}^{\tilde{M}_c} \frac{dN}{d\tilde{M}} d\tilde{M} \quad (3.62)$$

Equation (3.62) is solved numerically for $\tilde{M}_c \in [\tilde{M}_{c,min} \text{ and } \tilde{M}_{c,max}]$ to build a table of $n(< \tilde{M}_c)$ vs \tilde{M}_c values. For as many objects as are desired, a random number, \mathcal{R}_1 , is drawn, and $n(< \tilde{M}_c) = \mathcal{R}_1 n(< \tilde{M}_{c,max})$ is solved for the corresponding \tilde{M}_c using interpolation on the tabulated values. Every object has a different \mathcal{R}_1 , and so the solution \tilde{M}_c assigned as the mass of each object is also different. Only three types of initial mass function are used, a Schechter (1976) function (equation 2.2, given again here)

$$\frac{dN}{d\tilde{M}} \propto \tilde{M}^{-\beta_{MF}} \exp\left(-\frac{\tilde{M}}{\tilde{M}_{sch}}\right) \quad (3.63)$$

where β_{MF} controls the power-law component of the distribution, and \tilde{M}_{sch} controls where the exponential drop-off in the distribution becomes significant. Additionally a power-law IMF may be used, but this is just a Schechter (1976) function with

$\widetilde{M}_{sch} \rightarrow \infty$. The other initial tracer mass function available is a log-normal:

$$\frac{dN}{d \log \widetilde{M}} \propto \exp \left(- \frac{\left[\log \widetilde{M}_c - \widetilde{M}_c \right]^2}{2 \sigma_{\widetilde{M}_c}^2} \right) \quad (3.64)$$

where μ_M controls the mean of the distribution, and σ_M controls the dispersion.

To obtain the instantaneous radius of a tracer object, exactly the same idea is applied; the number of tracer objects interior to radius \tilde{r} is built as a tabulated function of \tilde{r} between $\tilde{r}_{trace,min}$ and $\tilde{r}_{trace,max}$:

$$N(< \tilde{r}) = 4\pi \int_{\tilde{r}_{trace,min}}^{\tilde{r}} \tilde{r}'^2 \tilde{\nu}(\tilde{r}') d\tilde{r}' \quad (3.65)$$

For each object, a random number, \mathcal{R}_2 , is drawn, $N(< \tilde{r}) = \mathcal{R}_2 N(< \tilde{r}_{max})$ is solved for \tilde{r} using interpolation, and the solution \tilde{r} , now denoted as \tilde{r}_M , is assigned as the initial instantaneous radius. Once \tilde{r}_M is known, $\tilde{\Phi} = \tilde{\Phi}(\tilde{r}_M)$ immediately follows from the output of Poisson's equation (equation 3.13).

From equation (3.50), the joint probability distribution of \tilde{E} and h at a fixed \tilde{r}_M is:

$$\mathcal{P}(\tilde{E}, h) = \frac{4\pi}{\tilde{r}_M \tilde{\nu}(\tilde{\Phi})} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right] \frac{hj(h)}{\sqrt{h_{max}^2 - h^2}} \quad (3.66)$$

Because of the quasi-separable nature of the distribution function, the orbital energy and orbital angular momentum can be sampled quasi-separately, i.e. sampling for \tilde{E} does not depend on h , whereas sampling for h does depend on \tilde{E} (through $h_{max}(\tilde{E}, \tilde{\Phi})$, cf. equation 3.49). Therefore the energy sampling is performed first, and the angular momentum sampling second. The appropriate functions for the respective samplings are:

$$\begin{aligned} \mathcal{P}(\tilde{E}) &= \frac{4\pi}{\tilde{r}_M \tilde{\nu}(\tilde{\Phi})} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right] \int_0^{h_{max}} \frac{hj(h)}{\sqrt{h_{max}^2 - h^2}} dh \\ \mathcal{P}(h) &= \frac{hj(h)}{\sqrt{h_{max}^2 - h^2}} \end{aligned}$$

the cumulative energy probability distribution is then:

$$\mathcal{P}(< \tilde{E}) = \frac{4\pi}{\tilde{r}_M \tilde{\nu}(\tilde{\Phi})} \int_{\tilde{\Phi}}^{\tilde{E}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right] \int_0^{h_{max}} \frac{hj(h)}{\sqrt{h_{max}^2 - h^2}} dh \quad (3.67)$$

such that integrating up to $\tilde{E} = \tilde{E}_{max}$ gives $\mathcal{P}(< \tilde{E}_{max}) = 1$ as of course it must. Equation (3.67) is then calculated numerically over $\tilde{E} \in [\tilde{\Phi}, \tilde{E}_{max}]$ to build an interpolable table. For each object, another random number, \mathcal{R}_3 , is drawn and $\mathcal{P}(< \tilde{E}) = \mathcal{R}_3 \mathcal{P}(< \tilde{E}_{max})$ is solved for \tilde{E} , which is assigned as the orbital energy. With \tilde{r} , $\tilde{\Phi}$ and \tilde{E} known, h_{max} can be calculated using equation (3.49). The orbital circularity of each object, h , is then assigned using the rejection method (e.g. Press et al. 1992) requiring a comparison function, $\mathcal{P}_{comp}(h)$ (van den Bosch et al. 1999):

$$\mathcal{P}_{comp}(h) = \frac{h_{max}}{\sqrt{h_{max}^2 - h^2}} \quad (3.68)$$

chosen such that $\mathcal{P}_{comp}(h) \geq \mathcal{P}(h)$ for all h (subject to the constraint that $j(h) \leq 1$) and for which the normalised cumulative probability distribution is analytically calculable and invertible, given by:

$$\mathcal{P}_{comp}(< h) = \frac{2}{\pi} \arcsin\left(\frac{h}{h_{max}}\right) \quad (3.69)$$

As this cumulative function is analytically invertible, an interpolation table is not necessary, but the general idea is the same — a random number, \mathcal{R}_4 , is drawn, and a random value of h is assigned: $h_{try} = h_{max} \sin\left(\frac{\mathcal{R}_4 \pi}{2}\right)$. However, this is based on the comparison function rather than the desired probability distribution. As the comparison function is everywhere greater than the desired probability distribution, another random number, \mathcal{R}_5 , is drawn and h_{try} is rejected if the comparison function evaluated at h_{try} multiplied by \mathcal{R}_5 is greater than the desired probability distribution evaluated at h_{try} , i.e. rejected if $\mathcal{R}_5 > \mathcal{P}(h_{try})/\mathcal{P}_{comp}(h_{try}) = hj(h)/h_{max}$, and therefore $j(h) \leq 1 \forall h$ is required for this to work. When h_{try} is rejected, a new \mathcal{R}_4 is drawn, and the procedure repeats until a value for h_{try} is accepted.

For constant anisotropy, $j(h) = h^{-2\beta_0}$ (cf. equation 3.57), and therefore $\beta_0 > 0$ means that $j(h) \rightarrow \infty$ as $h \rightarrow 0$, and therefore the rejection method cannot be used to populate a system with this circularity function. Fortunately, the integral over h in this case is:

$$\mathcal{P}(< h) = \int_0^h \frac{h^{1-2\beta_0}}{\sqrt{h_{max}^2 - h^2}} dh = \frac{h_{max}^{1-2\beta_0}}{2} \int_0^{h^2/h_{max}^2} u^{-\beta_0} (1-u)^{-1/2} du \quad (3.70)$$

which is just an incomplete β function, $B_{h^2/h_{max}^2}(1 - \beta_0, 1/2)$. Once normalised, this may be numerically integrated over $h^2/h_{max}^2 \in [0, 1]$, a point along the cumulative distribution assigned by a random number, and the integral inverted for h^2/h_{max}^2 in exactly the same way as the mass, energy and initial galactocentric position samplings.

With the addition of h and h_{max} , it is possible to calculate initial instantaneous values for \tilde{v}_r^2 and \tilde{v}_t^2 using equations (3.51) and (3.52). Another random number, \mathcal{R}_6 , is then drawn and used to determine the sign of \tilde{v}_r (i.e. positive if $\mathcal{R}_6 > 0.5$ and negative otherwise) in order to prevent any net flux. One more additional random number, \mathcal{R}_7 , is used to calculate $\gamma_{inc} = 2\pi\mathcal{R}_7$ to determine \tilde{v}_ϕ and \tilde{v}_θ using equations (3.45) and (3.46). As γ_{inc} is equally likely to be sampled in any $\pi/2$ quadrant, the signs of \tilde{v}_ϕ and \tilde{v}_θ are already randomised, and a further two random numbers to decide the signs of \tilde{v}_ϕ and \tilde{v}_θ are not necessary to prevent net rotation. Thus \tilde{v}_r , \tilde{v}_ϕ and \tilde{v}_θ are also known.

With \tilde{E} and h known, it is straightforward to calculate L using the definition of h (equation 3.2). With the addition of $\tilde{\Phi}$ known, it is possible to calculate the orbital pericentre and apocentre of this randomly generated object, by solving $\tilde{v}_r = 0$ (cf. equation 3.48) for \tilde{r} . This is simply:

$$2 \left[\tilde{E} - \tilde{\Phi}(\tilde{r}_{ap}) \right] \tilde{r}_{ap}^2 - \tilde{L}^2 = 0 \quad (3.71)$$

where \tilde{r}_{ap} is either the orbital pericentre, \tilde{r}_p , or the orbital apocentre, \tilde{r}_a . When $\tilde{L} = 0$, the two solutions are clearly $\tilde{r}_p = 0$ and $\tilde{r}_a = \tilde{r}(\tilde{\Phi} = \tilde{E})$, whereas $\tilde{L} = \tilde{L}_c(\tilde{E})$ corresponds to a circular orbit, and so the solutions are $\tilde{r}_p = \tilde{r}_a = \tilde{r}_c(\tilde{E})$. Both solutions are therefore always bracketed by:

$$\begin{aligned} 0 &\leq \tilde{r}_p \leq \tilde{r}_c(\tilde{E}) \\ \tilde{r}_c(\tilde{E}) &\leq \tilde{r}_a \leq \tilde{r}(\tilde{\Phi} = \tilde{E}) \end{aligned}$$

Therefore \tilde{r}_p and \tilde{r}_a are easily found by bisection. The ellipticity of the orbit in question is then simply:

$$e = \frac{\tilde{r}_a - \tilde{r}_p}{\tilde{r}_a + \tilde{r}_p} \quad (3.72)$$

The radial period is defined as:

$$\tilde{T}_r = 2 \int_{\tilde{t}(\tilde{r}_p)}^{\tilde{t}(\tilde{r}_a)} d\tilde{t} = 2 \int_{\psi(\tilde{r}_p)}^{\psi(\tilde{r}_a)} \frac{d\tilde{t}}{d\psi} d\psi$$

where $\tilde{t}(\tilde{r}_{ap})$ is the time at pericentre/apocentre, and $\psi(\tilde{r}_{ap})$ is the azimuthal angle at pericentre/apocentre (measured from any arbitrary orbital phase, ψ_0). Furthermore, $d\psi/d\tilde{t} = \tilde{v}_t/\tilde{r} = \tilde{L}/\tilde{r}^2$, and thus the radial period may be written as:

$$\tilde{T}_r = 2 \int_{\psi(\tilde{r}_p)}^{\psi(\tilde{r}_a)} \frac{\tilde{r}^2(\psi)}{\tilde{L}} d\psi \quad (3.73)$$

While the azimuthal period is given by:

$$\tilde{T}_\psi = \int_{\psi_0}^{\psi_0+2\pi} \frac{\tilde{r}^2(\psi)}{\tilde{L}} d\psi \quad (3.74)$$

where ψ_0 is an arbitrary constant.

In order to acquire $\tilde{r}(\psi)$, the equation for radial motion of a particle is obtained by differentiating the equation for radial speed (equation 3.48) with respect to time, and substituting $\frac{d\psi}{d\tilde{t}} = \tilde{L}/\tilde{r}^2$ to obtain the following differential equation:

$$\frac{d^2\tilde{r}}{d\psi^2} = \tilde{r} - \frac{\tilde{r}^4}{\tilde{L}^2} \frac{d\tilde{\Phi}}{d\tilde{r}} + \frac{2}{\tilde{r}} \left(\frac{d\tilde{r}}{d\psi} \right)^2 \quad (3.75)$$

This differential equation is solved numerically over a range of $\psi \in [\psi_0, \psi_0 + 2\pi]$ to give $\tilde{r}(\psi)$. Maxima and minima of \tilde{r} as a function of ψ correspond to pericentric and apocentric passages, and thus $\psi(\tilde{r}_p)$ and $\psi(\tilde{r}_a)$ are also obtained. This information may then be used to calculate the radial and azimuthal periods (equations 3.73 and 3.74), in addition to the average orbital precession via the ratio of the radial and azimuthal periods. With the orbital periods of each tracer known, mass-loss rates due to tidal shocking may be calculated when evolving the GCS.

3.7 Simulated GCS set-up Code Description

All of this information has been compiled into a code (see Appendix E), which uses the equations and methods given above, along with user provided information, to create

a simulated GCS. The initial GCS code first prompts a selection of model family, i.e. polytropes, King models, Dehnen models, or power-laws, to describe the host galaxy. The next input is the parameter specifying which model of the selected family to use; k for polytropes or power-laws, γ for Dehnen models, or W_0 for King models (see section 3.3). Following this, a seed is required for the random number generation. With the host galaxy specified, the remaining input concerns the tracer GCS population. Firstly the tracer population number density profile must be specified, with the options of directly tracing the host galaxy (i.e. the tracer number density profile matches the density profile of the host galaxy) or double power-law profile (an $\alpha_{pl}\beta_{pl}\gamma_{pl}$ model; Zhao 1997), given by:

$$\tilde{\nu} = \tilde{r}^{-\gamma_{pl}} \left(1 + \tilde{r}^{1/\alpha_{pl}}\right)^{(\gamma_{pl}-\beta_{pl})\alpha_{pl}} \quad (3.76)$$

If the double power-law profile is selected, the logarithmic inner slope (γ_{pl}), the power-law outer slope (β_{pl}), and width of the transition region (α_{pl}) must subsequently be entered. Next the user is prompted to select the tracer CIMF, with the options of Schechter (1976), a Gaussian, or a power-law. If a Schechter (1976) CIMF is selected, the next information to be entered is the exponential cut-off mass, M_{sch} , then the logarithmic slope of the power-law component, β_{MF} (cf. equation 2.2). If a power-law CIMF is selected, only the logarithmic slope of the power-law component need be entered. Otherwise if a Gaussian CIMF is selected, the mean of the distribution, μ_M , and the CIMF dispersion, σ_M , must be selected (cf. equation 3.64). Finally, the maximum and minimum initial galactocentric radii allowed to be assigned to a tracer are specified, such that every tracer in the system will initially lie between the selected radii. Note that this does not forbid tracers from moving to radii outside of these limits during the time-evolution phase however, as this will depend on the assigned orbital energy and angular momentum. The point of specifying the maximum and minimum initial galactocentric radii is to prevent the majority of sampled tracers being placed at unrealistically small radii in models with central singularities, or out at unrealistically large radii in models with formally infinite total mass, i.e. to mimic real systems as

closely as possible.

Once all of the host system parameters and tracer initial properties have been calculated, the GCS is evolved through time with a second code. The evolution code reads in all of the data about the host galaxy, such as density, gravitational potential, derivatives of gravitational potential, velocity dispersions, etc. Furthermore, the tracer distribution function data is read in, and moreover the initial parameters of each tracer, such as orbital energies and angular momenta, radial and tangential speeds, galactocentric radii, and masses are read in. One by one each tracer is then evolved using a user specified combination of prescriptions taken from the literature and Chapter 2 (see section 5.2). All of these prescriptions are divided into one of four types; evaporation, shocks, stellar evolution, and dynamical friction. The code allows any one (or none) prescription from each type for a custom evolutionary recipe.

Thus the orbital parameters required for calculating subsequent evolution are obtained for as many tracers as are desired in the tracer system, for a general tracer density profile, $\tilde{\nu}$, with a general circularity function, $j(h)$, and with a general tracer mass function, $dN/d\log \widetilde{M}$, in a wide range of possible host galaxy systems. Demonstrations of the application of this method, followed by a detailed discussion of the dependence of velocity anisotropy on the circularity function, will be given in Chapter 4. A description of the procedure for taking an initial tracer system as produced using the method outlined in this Chapter, and evolving it through time will then be given in Chapter 5. This will then be followed by demonstrations of the application of this time-evolution procedure in Chapter 6.

4 Examples of Simulated Initial GCSs

4.1 Method Verification

Since the method for randomly sampling objects is designed to reproduce a continuous distribution with a discrete number of points, it is simple to verify whether this has worked as intended or not by overlaying the appropriately normalised continuous function and discrete data points. This is done first for the simplest aspect of the sampling, the CIMFs. The sampling for instantaneous orbital radius, orbital energy, and orbital radial and tangential speeds however are much more complicated, as they are interrelated via $\tilde{E} = \tilde{\Phi}(\tilde{r}_M) + \frac{1}{2}(\tilde{v}_r^2 + \tilde{v}_t^2)$. Therefore the sampled data must simultaneously satisfy the expected energy, spatial, and kinematic distributions.

The instantaneous orbital radii samplings are tested by binning the sampled points in terms of radius, and calculating the number density in each bin. Both the binned sampled tracers and the numerically recovered number density (i.e. integrated from the distribution function solved for in equation 3.50) are normalised against the total number of tracers, i.e. such that:

$$\sum_{i=1}^{N_b} \mathcal{N}_i = 4\pi \int_{\tilde{r}_{trace,min}}^{\tilde{r}_{trace,max}} \tilde{r}^2 \nu(\tilde{r}) d\tilde{r} = 1 \quad (4.1)$$

where N_b is the number of radial bins, and \mathcal{N}_i is the number of tracers in the volume spanned by the i 'th bin. This then allows the curve and the data to be overlayed.

In a similar fashion, the orbital energy samplings are tested by binning the sampled points in terms of orbital energy to give the number per unit energy. The expected number of objects per unit energy in a system is called the differential energy distribution, and is given by (see Appendix D):

$$\frac{dN}{d \log \tilde{E}} = (4\pi)^2 \ln(10) \tilde{E} \left[\tilde{L}_c(\tilde{E}) + \tilde{L}_0 \right] \tilde{g}(\tilde{E}) \int_{\tilde{r}_1}^{\tilde{r}_2} d\tilde{r} \tilde{r} \int_0^{h_{max}} \frac{h j(h) dh}{\sqrt{h_{max}^2 - h^2}} \quad (4.2)$$

where $\tilde{r}_1 = \tilde{r}_{min,trace}$ and $\tilde{r}_2 = \min[\tilde{r}_{max,trace}, \tilde{r}(\tilde{\Phi} = \tilde{E})]$. Both the binned sampled points and the curve are then normalised against the total number of tracers, such that

$$\sum_{i=1}^{N_b} \mathcal{N}_i = \int_{\log \tilde{\Phi}(\tilde{r}_1)}^{\log \tilde{E}_{max}} \frac{dN}{d \log \tilde{E}} d \log \tilde{E} = 1 \quad (4.3)$$

with $dN/d \log \tilde{E}$ given by equation (4.2), and $\tilde{E}_{max} = \tilde{\Phi}(\tilde{r}_f)$ being the maximum energy attainable (i.e. just enough for a tracer to reach the outermost radius allowed, \tilde{r}_f). In this case \mathcal{N}_i is the number of tracers with energies inside of those spanned by the i 'th bin.

The kinematic sampling is slightly different, in that rather than comparing number densities as with the instantaneous orbital radius and orbital energy samplings, the statistics of the distribution are compared instead, namely the velocity dispersions. Each sampled tracer is assigned two components of velocity; radial (\tilde{v}_r), and tangential (\tilde{v}_t). The tangential component is then broken down further into polar (\tilde{v}_θ) and azimuthal (\tilde{v}_ψ) components, with $\tilde{v}_\theta = \tilde{v}_t \cos \gamma_{inc}$ and $\tilde{v}_\psi = \tilde{v}_t \sin \gamma_{inc}$ where γ_{inc} is the orbital inclination, such that $\tilde{v}_t^2 = \tilde{v}_\theta^2 + \tilde{v}_\psi^2$. Since the system is assumed to be spherical (i.e. γ_{inc} is equally likely to take any value within $[0, 2\pi]$), this implies for the velocity dispersions that $\tilde{\sigma}_t^2 = \tilde{\sigma}_\theta^2 + \tilde{\sigma}_\psi^2$. Using this, it is then possible to bin tracers radially and calculate the standard deviations of each component of velocity, for comparison to the curves predicting radial and tangential velocity dispersions as a function of radius given by equations (3.54 and 3.55). This then facilitates comparison of the velocity anisotropy profile, using equation (3.41). Moreover, while the standard deviations of each component of velocity in each bin must match the predicted curves, the averages of the velocity components in each bin must equate to zero to satisfy the requirement that net flux and net rotation are zero.

Beginning with the simplest case, sampled tracer masses are binned and plotted against the function governing their distribution. As with both the initial orbital radius and the orbital energy samplings, both the sampled data and the CIMF are normalised against the total number of tracers, i.e.:

$$\sum_{i=1}^{N_b} \mathcal{N}_i = \int_{\log \widetilde{M}_{c,min}}^{\log \widetilde{M}_{c,max}} \frac{dN}{d \log \widetilde{M}} d \log \widetilde{M} = 1 \quad (4.4)$$

where N_b is the number of mass bins, and \mathcal{N}_i is the number of tracers with masses spanned by the i 'th bin. This is displayed for a Schechter (1976) CIMF (equation 2.2) in Figure 4.1, and for a Gaussian CIMF (equation 3.64) in Figure 4.2.

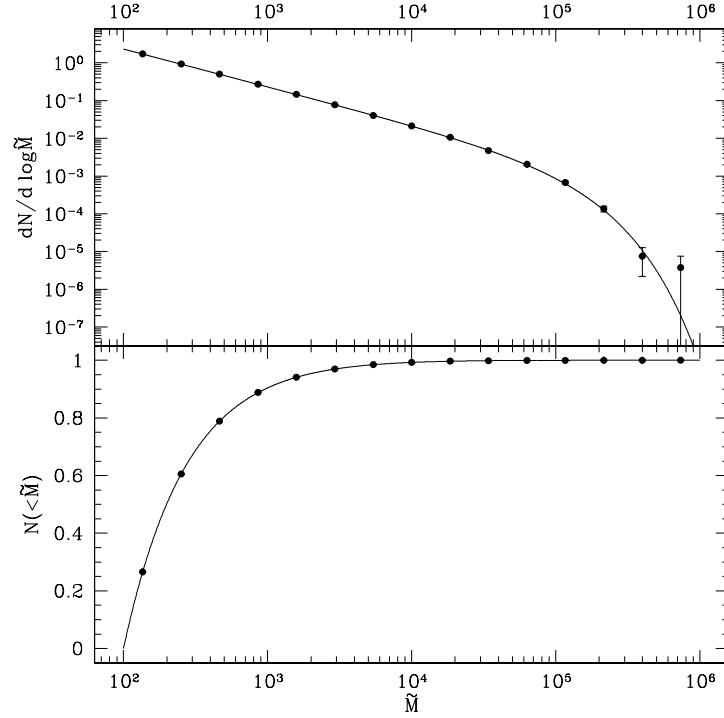


Figure 4.1: Comparison of continuous Schechter (1976) function with $\beta_{MF} = 2$ and $\widetilde{M}_{sch} = 10^5$ (cf. equation 2.2) and appropriately binned sampled tracer masses. In both panels the solid curve is the continuous function and the points are the binned sampled tracers. Top panel: the logarithmic mass range is divided into equal width bins, and each of the 10^6 sampled tracers are placed into their respective appropriate mass bin. Bottom panel: for each bin, the total number of sampled tracers with masses equal to or less than the current bin mass are totalled. In both panels, the curve and the bins are normalised according to equation (4.4).

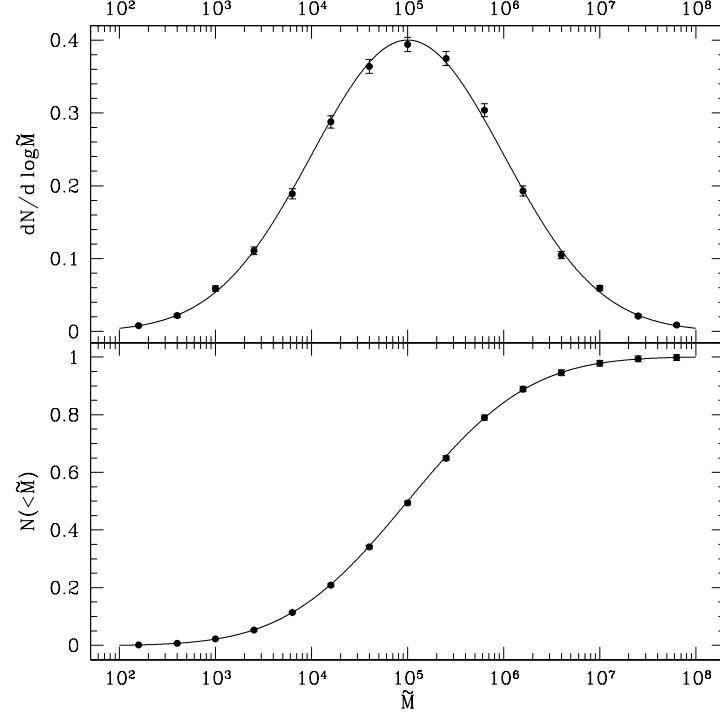


Figure 4.2: Comparison of continuous Gaussian function with $\mu_M = 5$ and $\tilde{\sigma}_M = 1$ (cf. equation 3.64) and appropriately binned sampled tracer masses. In both panels the solid curve is the continuous function and the points are the binned sampled tracers. Top panel: the logarithmic mass range is divided into equal width bins, and each of the 10^4 sampled tracers are placed into their respective appropriate mass bin. Bottom panel: for each bin, the total number of sampled tracers with masses equal to or less than the current bin mass are totalled. In both panels, the curve and the bins are normalised according to equation (4.4).

Next are displayed sets of plots depicting the properties of a range of sampled GCSs in different host galaxies, with different spatial and velocity distributions via $j(h)$ and \tilde{L}_0 . The top left plot in each set depicts the number density, $\tilde{\nu} = \frac{dN}{dV}$, as calculated with equation (3.50) with bins of appropriately normalised sampled tracers, and displays the reduced chi-squared to indicate how well the binned sampled tracers match the curve. Additionally, the original analytical input tracer number density is overlaid (i.e. one of equations 3.16, 3.25, 3.28, 3.31 for spatial profiles directly tracing the host galaxy, or 3.76 for double power-law tracer spatial profiles). The bottom

left panel displays the difference between the number density, $\tilde{\nu}$, and the analytical input tracer number density, $\tilde{\nu}_0$, as a function of galactocentric distance. The top right panel displays the differential energy distribution as calculated using equation (4.2), with appropriately normalised bins of sampled tracer orbital energies. Furthermore, each differential energy distribution plot displays the reduced chi-squared to indicate how well the binned sampled tracers match the calculated curve. The bottom right plot depicts three panels: the bottom panel displays curves of the radial and tangential velocity dispersions as calculated with equations (3.54) and (3.55), and is overlaid with the radial (red points), and tangential ($\tilde{\sigma}_t^2 = \tilde{\sigma}_\theta^2 + \tilde{\sigma}_\psi^2$, black points) velocity dispersions of tracers in each bin. Furthermore, the radial and tangential velocity dispersions each have reduced chi-squared values indicating how well the binned points match the curves. The middle panel displays the velocity anisotropy profile as calculated with equation (3.56) and is overlaid with the velocity anisotropy in each tracer bin as calculated with equation (3.41). The top panel displays the average tracer radial, polar, and azimuthal velocities in each bin, and for no significant net flux or rotation of the GCS must be near zero. Since in general the differential energy distribution and velocity dispersions are calculated from the distribution function (cf. equations 3.54, 3.55, and 4.2) which is solved for with the use of numerical quadrature (see section 3.4.1), unless the GCS is isotropic there are no analytic curves to compare them to as can be done with the number density (in fact, even with isotropy sometimes the differential energy distribution and velocity dispersion are non-analytic, e.g. Dehnen 1993).

First are displayed the initial orbital radii, orbital energies, and kinematic distributions of an isotropic simulated GCS with $\tilde{L}_0 = 0$ that directly traces the density distribution of the host galaxy. These initial GCS models are displayed for a range of Dehnen $\gamma = 0, 1, 2$ host galaxies in Figures 4.3, 4.4, 4.5, King $W_0 = 10, 15, 20$ host galaxies in Figures 4.6, 4.7, 4.8, polytrope $k = 1.5, 2, 2.5$ host galaxies in Figures 4.9, 4.10, 4.11, and power-law $k = 1.5, 2, 2.5$ host galaxies in Figures 4.12, 4.13, 4.14.

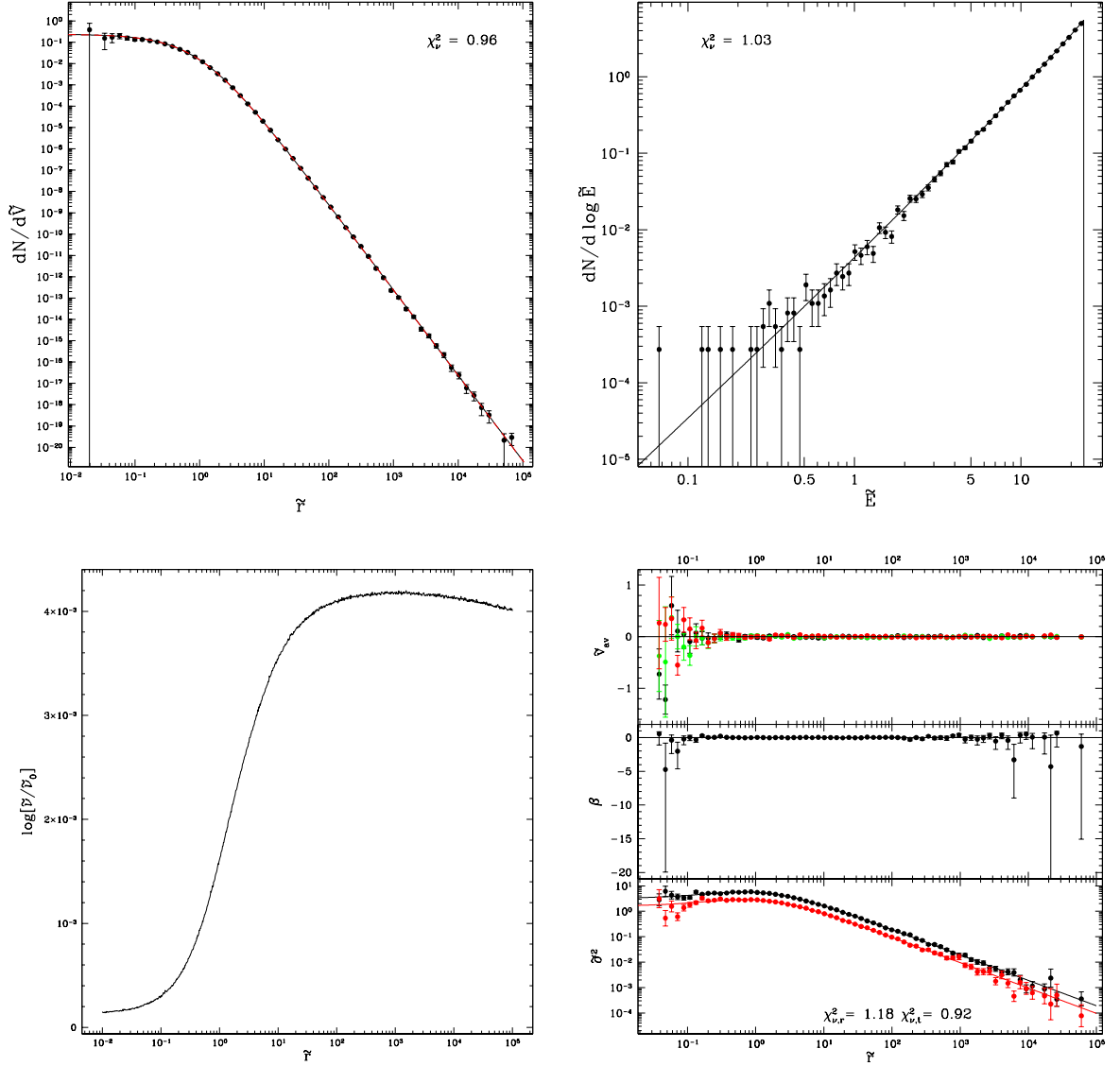


Figure 4.3: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $\gamma = 0$ Dehnen model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

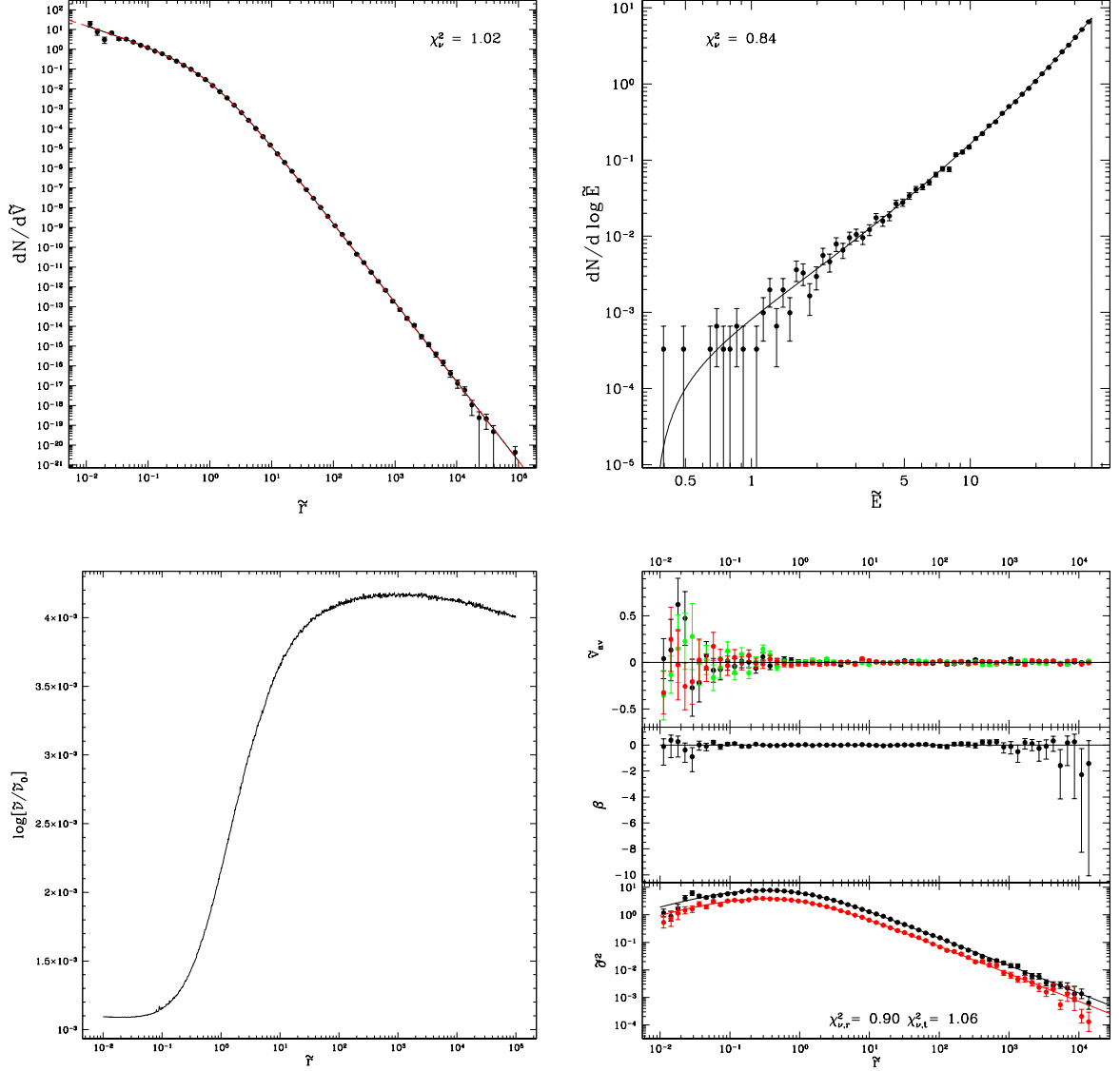


Figure 4.4: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $\gamma = 1$ Dehnen model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

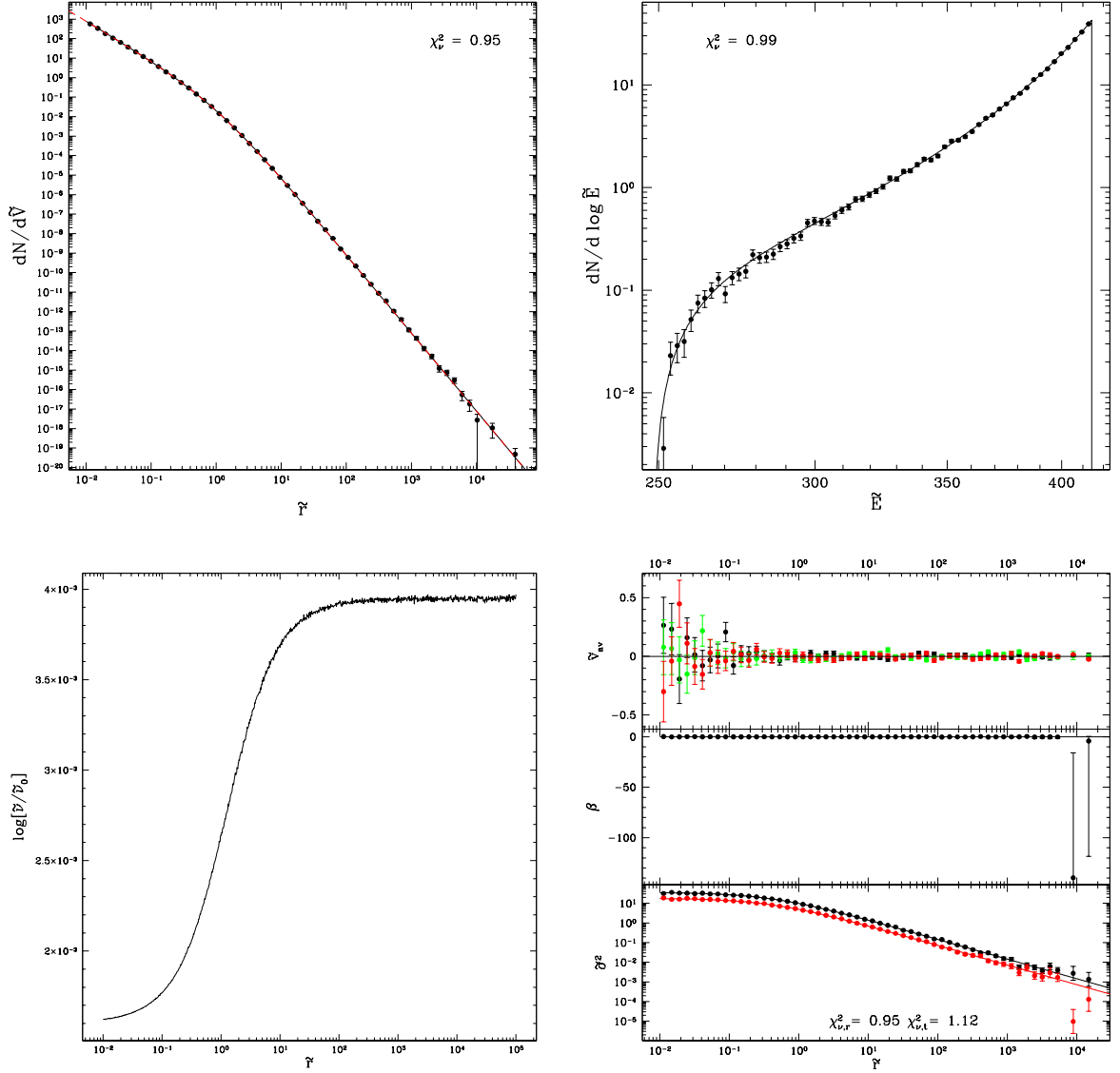


Figure 4.5: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $\gamma = 2$ Dehnen model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

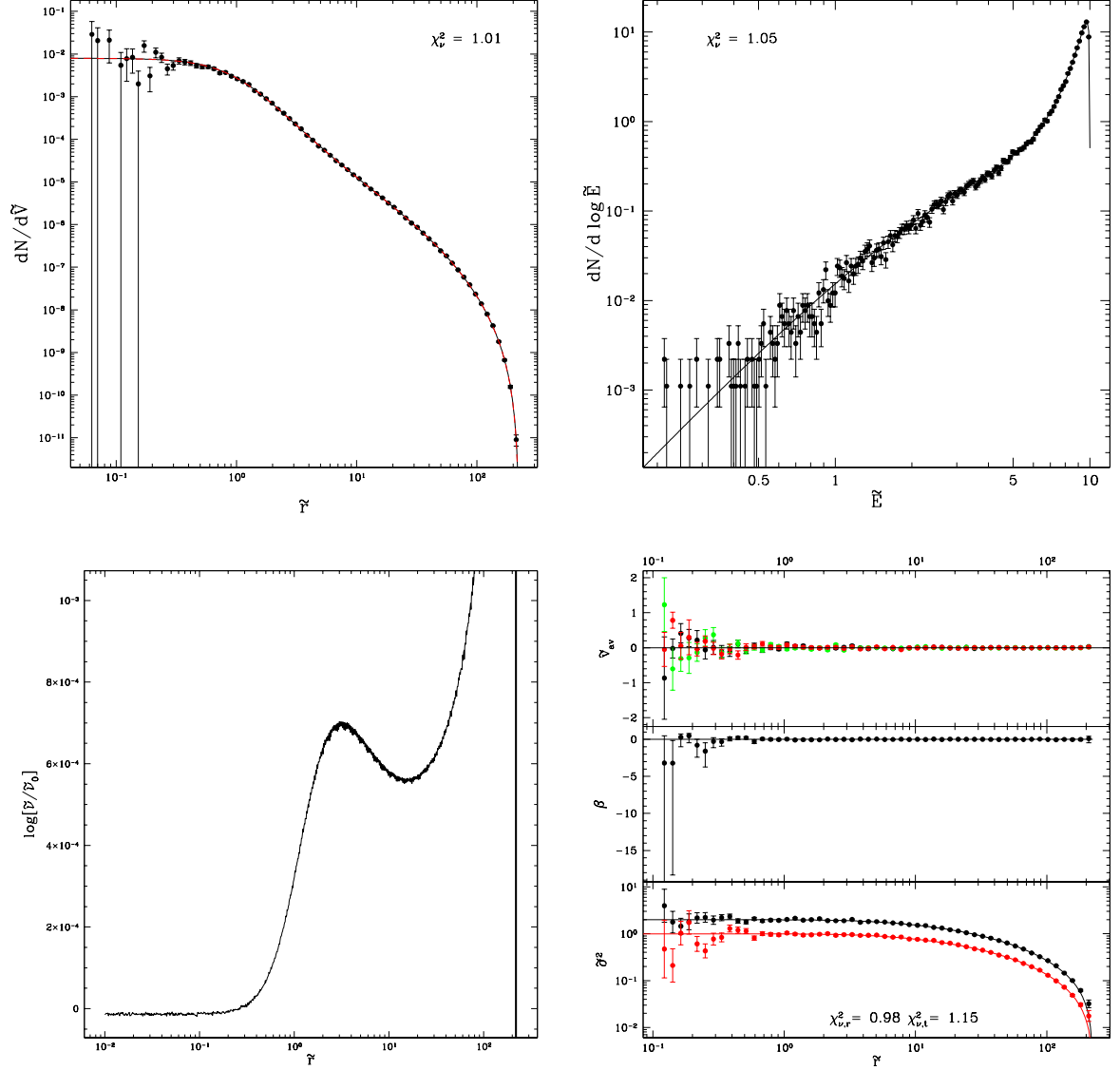


Figure 4.6: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $W_0 = 10$ King model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

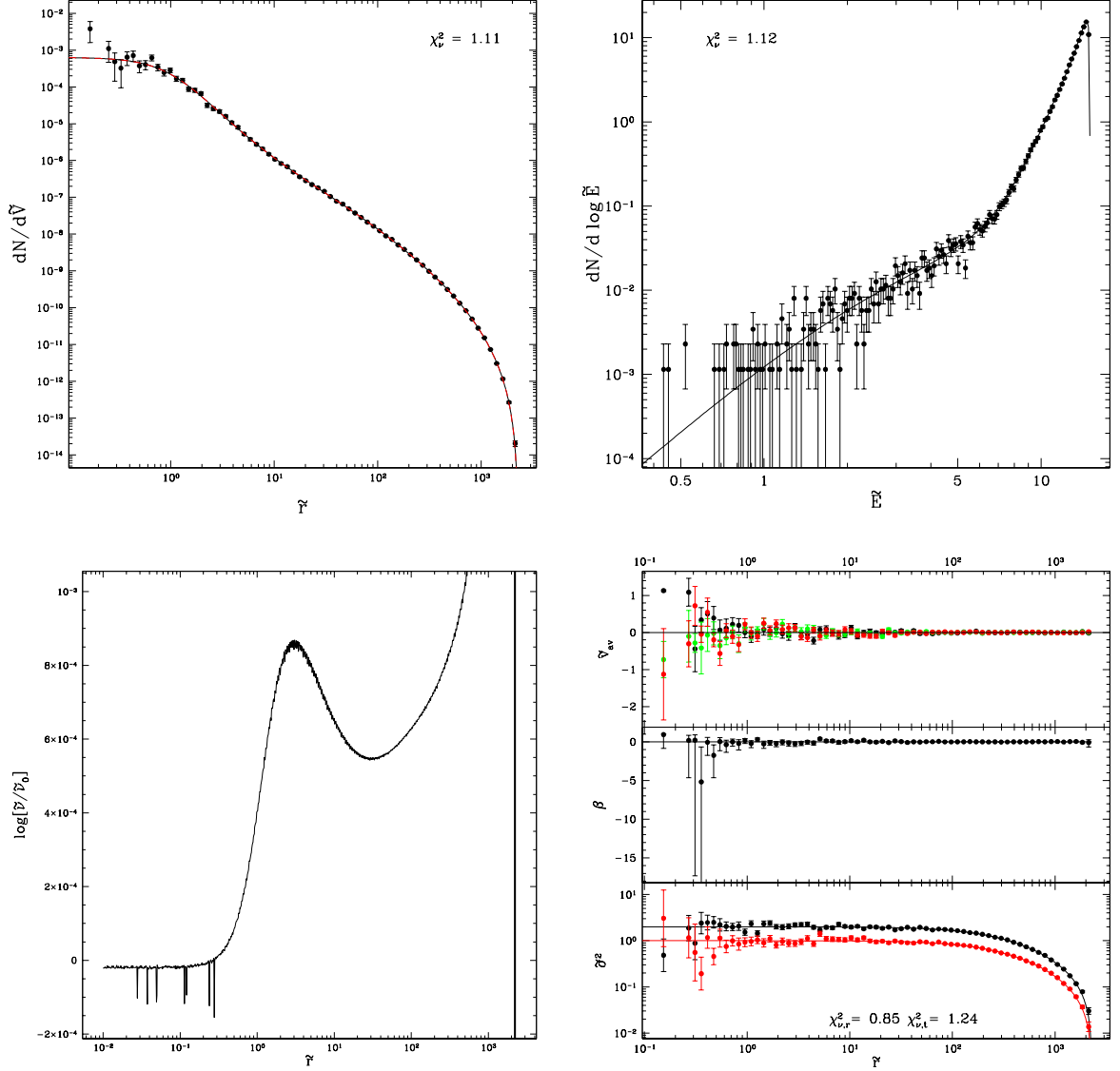


Figure 4.7: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $W_0 = 15$ King model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

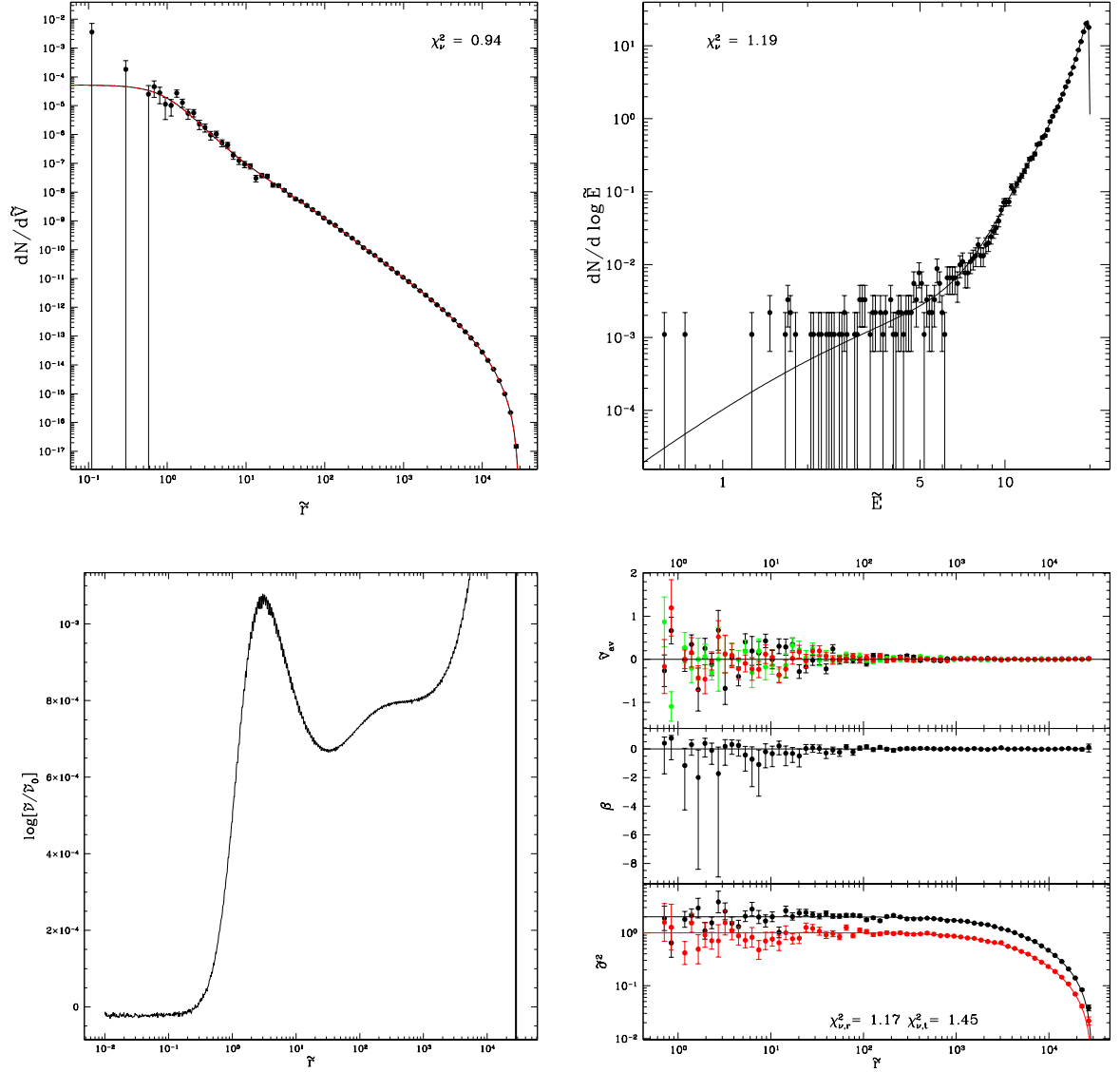


Figure 4.8: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $W_0 = 20$ King model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

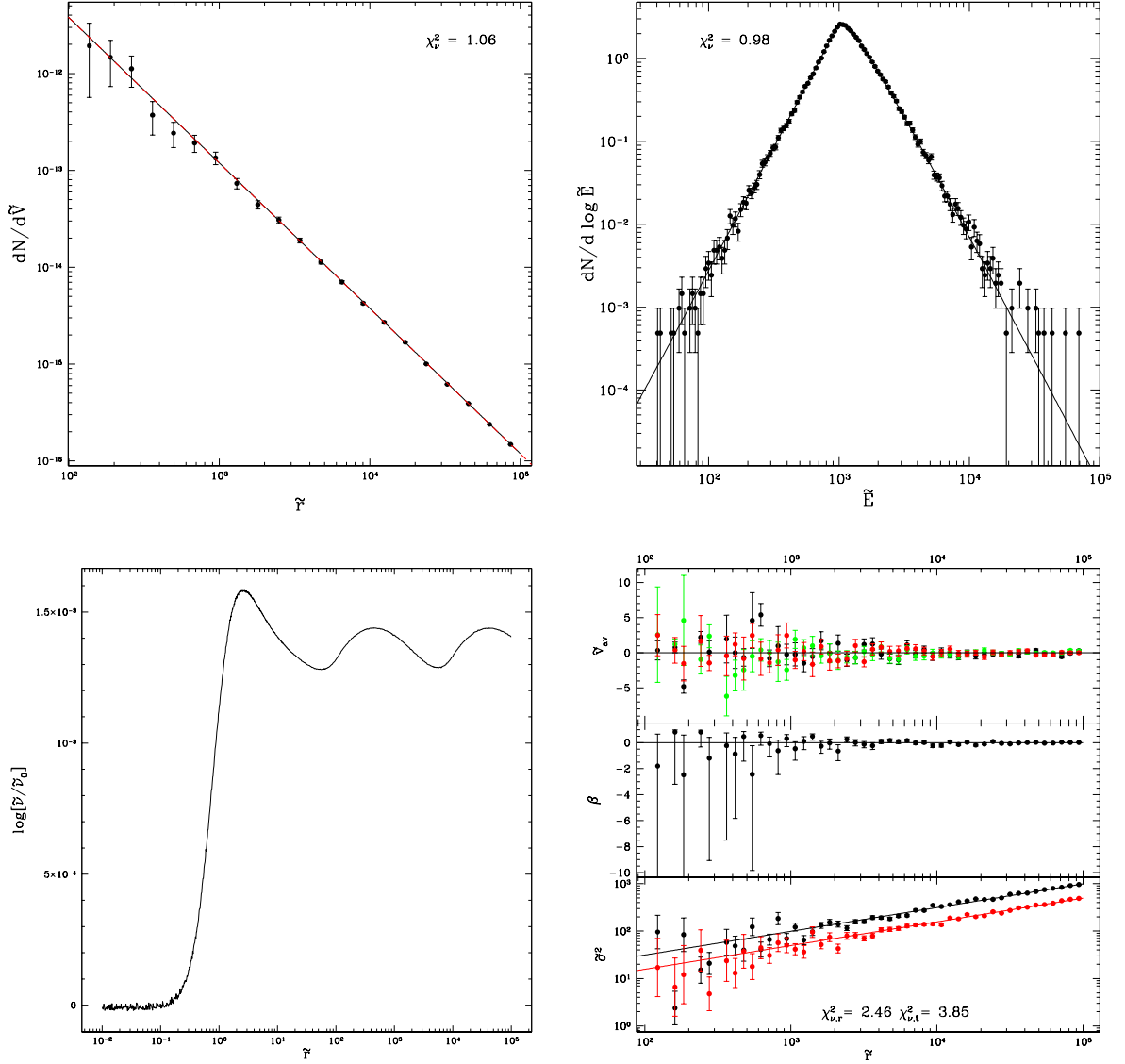


Figure 4.9: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $k = 1.5$ polytrope model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

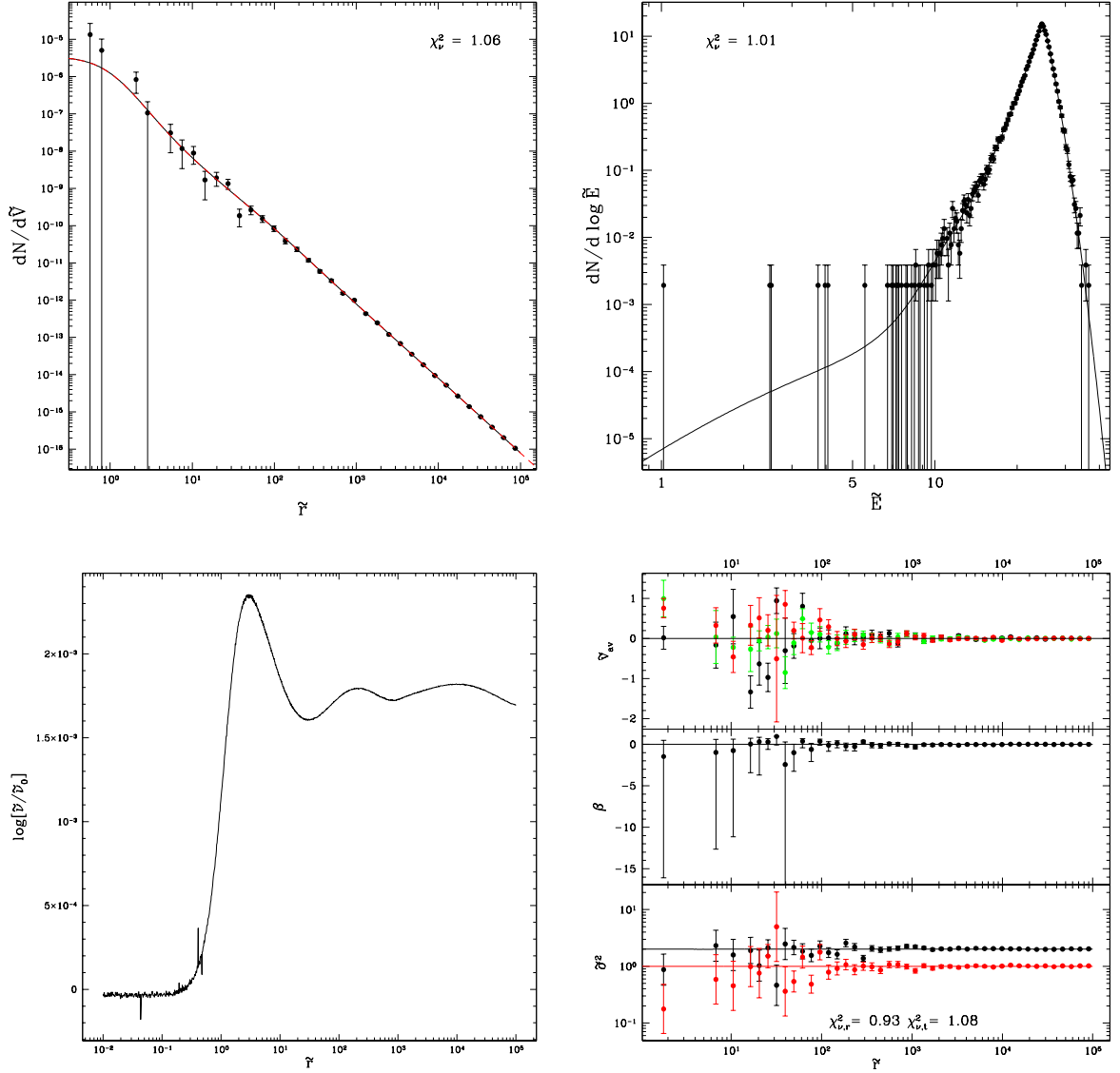


Figure 4.10: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $k = 2$ polytrope model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

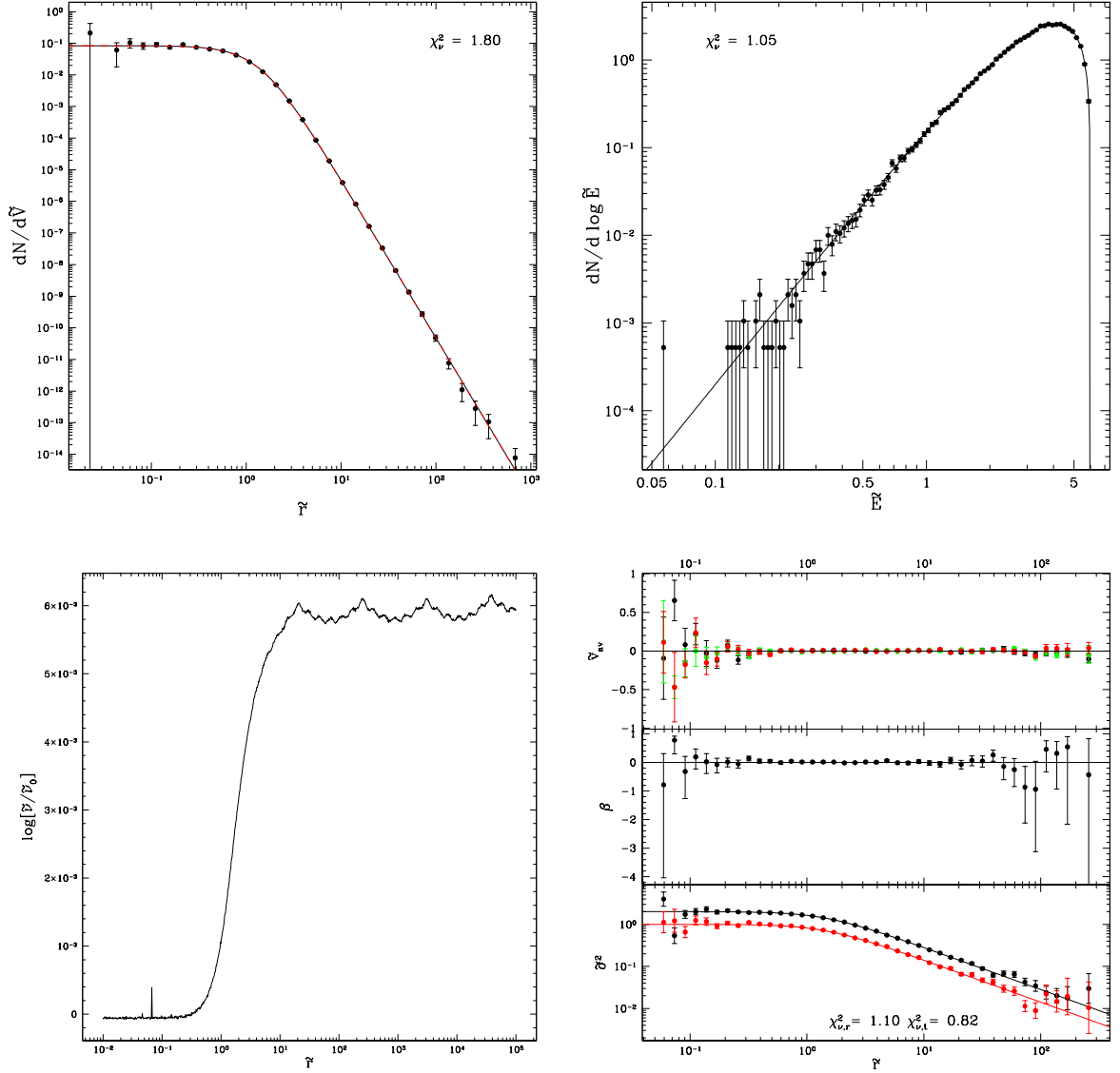


Figure 4.11: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $k = 2.5$ polytrope model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

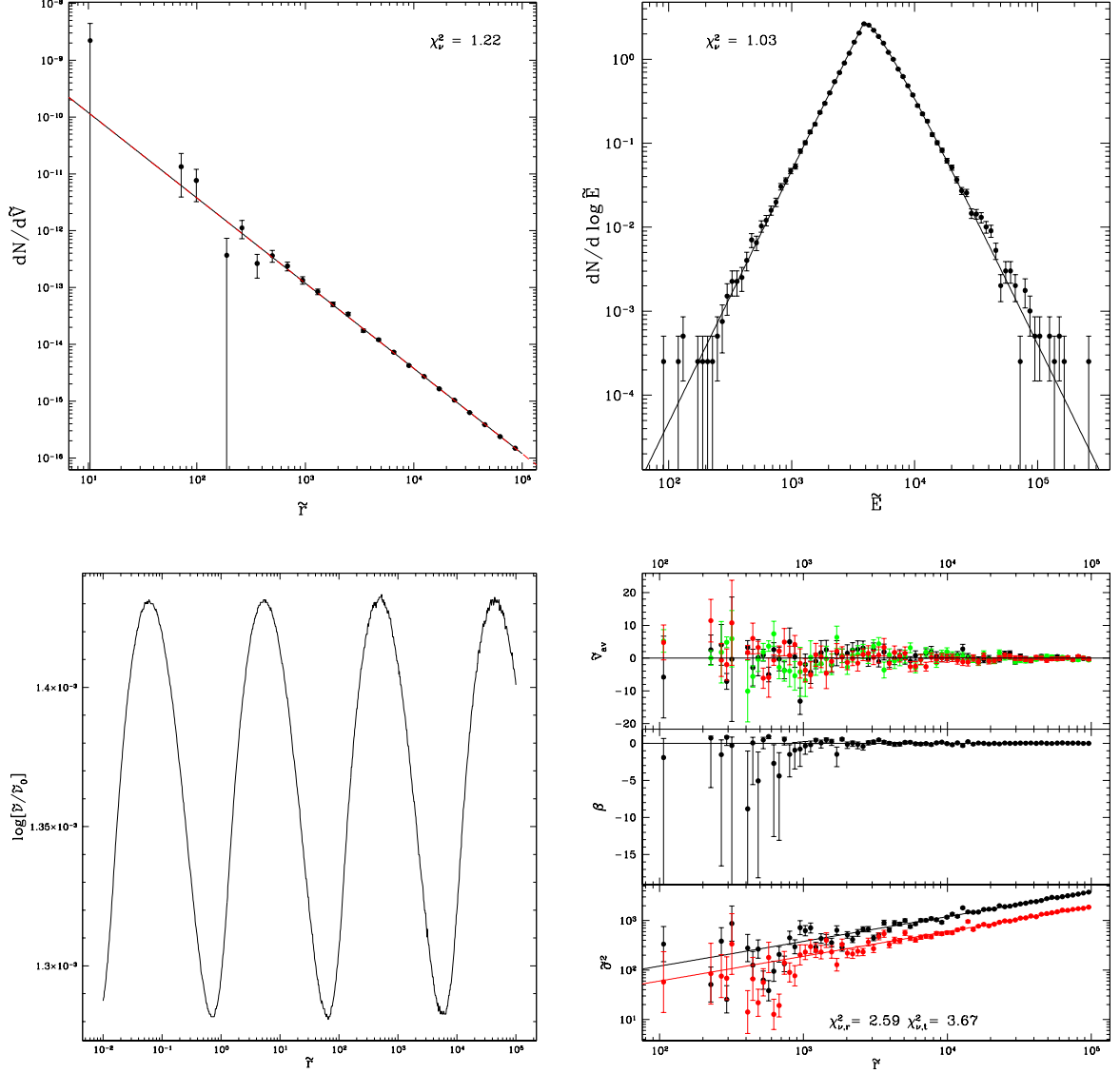


Figure 4.12: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $k = 1.5$ power-law model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

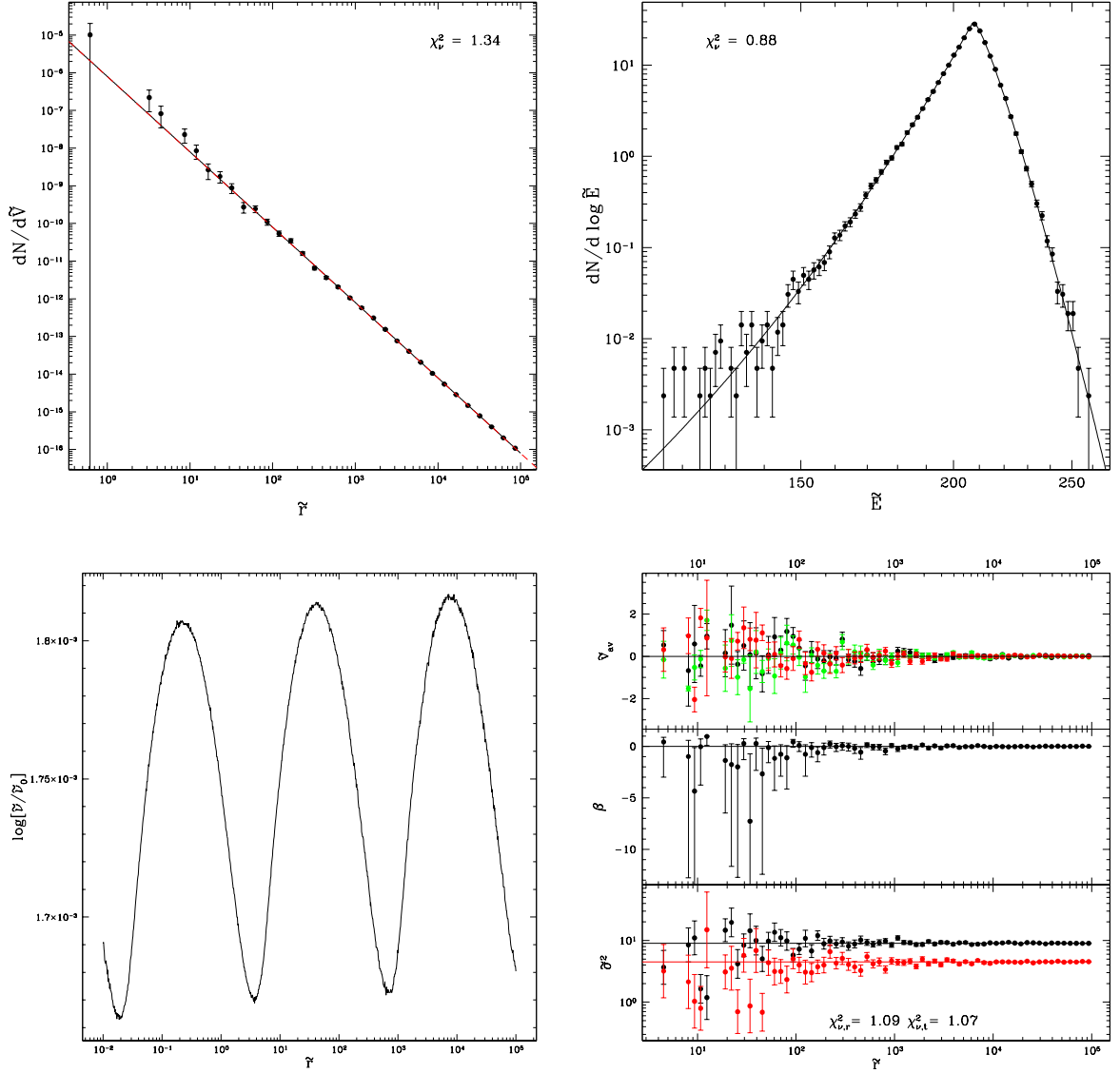


Figure 4.13: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $k = 2.0$ power-law model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

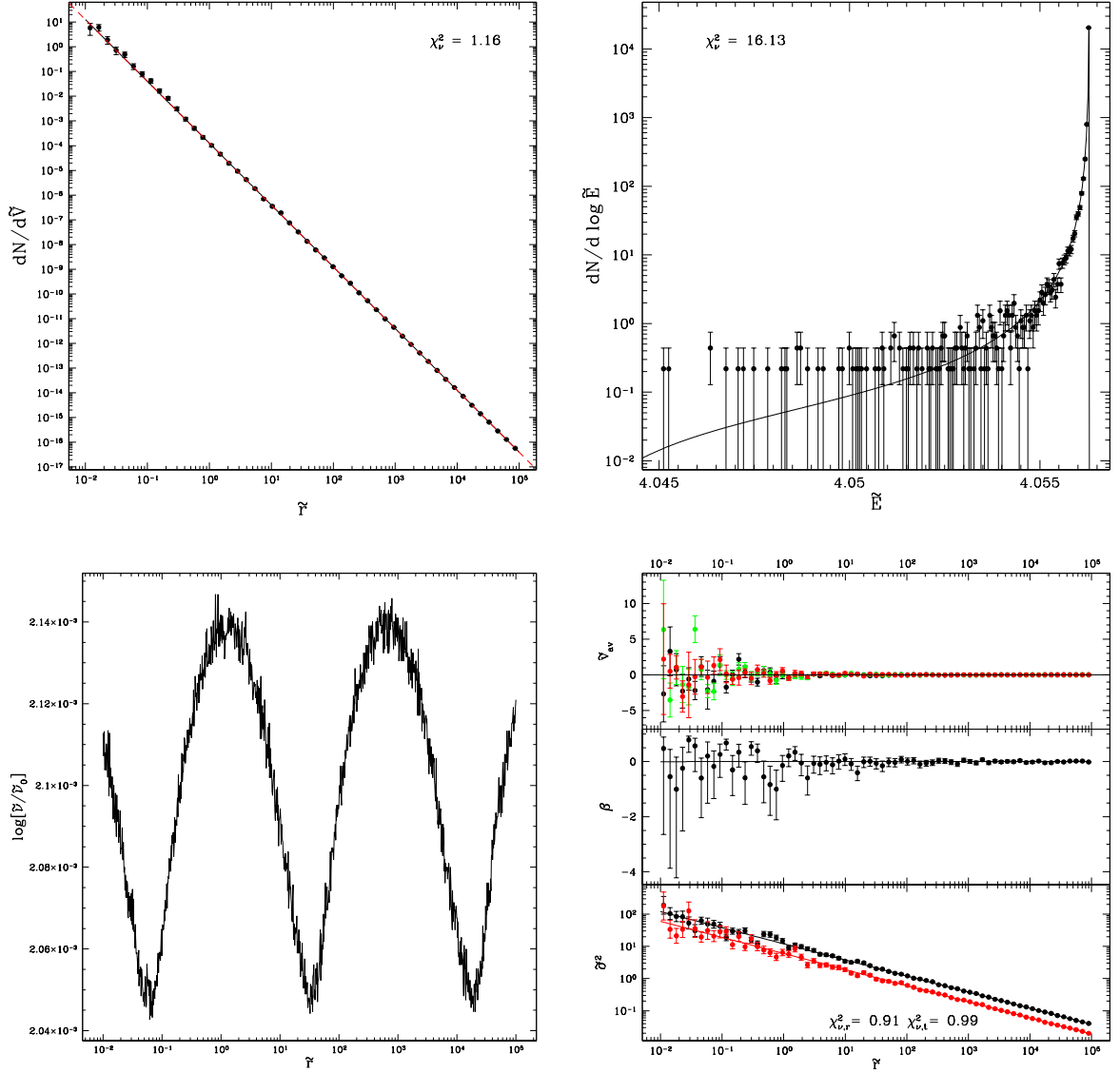


Figure 4.14: Properties of an isotropic GCS with $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $k = 2.5$ power-law model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

Clearly in the very simple case of spatial distributions that directly trace the host galaxy and isotropic velocity distributions, the sampling has generally worked very well. The residuals of the number density curves (bottom left plot) indicate that all the King models over-predict the number density near the zero-density radius, however this is simply a manifestation of the difficulty of matching together two functions that are plummeting to negative infinity. Furthermore, the differential energy distribution of the $k = 2.5$ power-law (Figure 4.14) has a relatively high reduced chi-squared of $\chi^2_\nu = 16.01$. However, essentially all of the ‘badness’ in the reduced chi-squared comes from the rightmost bin, and is simply a result of fitting rectangular bins to a very steeply increasing curve — increasing the number of bins would address this issue, but doing so makes the bins at lower energies underpopulated, dramatically increasing the noise. Inspection of the plot clearly shows that the curve and binned tracers are actually in very close agreement for the most part. Note that by limiting the energy that tracers can be sampled at to \tilde{E}_{max} , the radius corresponding to this energy ($\tilde{r}_f[\tilde{\Phi} = \tilde{E}_{max}]$) may only be reached by purely radial orbits. This is because a non-zero angular momentum corresponds to a non-zero tangential speed at apocentre ($\tilde{v}_t = \tilde{L}/\tilde{r}_a$), and so the particle must also have non-zero kinetic energy at apocentre. Therefore not all orbital energy can be in the form of gravitational potential, and so $\tilde{r}_a \neq \tilde{r}_f(\tilde{\Phi} = \tilde{E}_{max})$ despite having the maximum orbital energy attainable. Therefore, orbits of all angular momenta are allowed up to the radius of a circular orbit with $\tilde{r}_c(\tilde{E}_{max})$. However, radii beyond this are only accessible to increasingly eccentric orbits, to the extreme of only zero angular momentum orbits at \tilde{r}_f . Thus, between $\tilde{r}_c(\tilde{E}_{max})$ and \tilde{r}_f the differential energy distribution must decrease, which is why the curve dives down almost vertically at the end. All of the remaining sets of plots have reduced chi-squared values near to 1, indicating that the sampling is working very well, and the residuals indicate an error of less than 1% (apart from near the zero-density radius in King host galaxy models), showing that the numerical solution of the distribution function is also working very well. Furthermore, apart from the $k = 1.5$ polytrope and power-law host galaxy models in Figures 4.9 and 4.12, the radial and tangential velocity dispersion reduced chi-squared values are satisfactorily near to unity. The reason the reduced chi-squared

is somewhat higher in these two models is simply because the relatively shallow density gradient means that the vast majority of tracers are sampled out at large radii (cf. the spatial sampling plots — virtually no tracers were sampled inside of $\tilde{r} < 10^2$ in either case), and consequently the inner regions are very noisy, which drives up the reduced chi-squared.

Next, an additional layer of complication is added by moving away from isotropy, to constant tangential anisotropy (equation 3.57, $j(h) = h^{-2\beta_0}$ with $\beta_0 = -1$). The other properties of the simulated GCSs remain the same however (i.e. spatial profiles directly tracing the host galaxy, and $\tilde{L}_0 = 0$). Rather than display these GCSs for several models in each of the four possible host galaxies, they are displayed for only one model in each family, namely: Dehnen $\gamma = 1$ (Hernquist 1990 model); King $W_0 = 20$; polytrope $k = 2.5$ (Plummer 1911 Sphere); and power-law $k = 2$ (Singular Isothermal Sphere) host galaxies. These are displayed in Figures 4.15, 4.16, 4.17, 4.18 respectively.

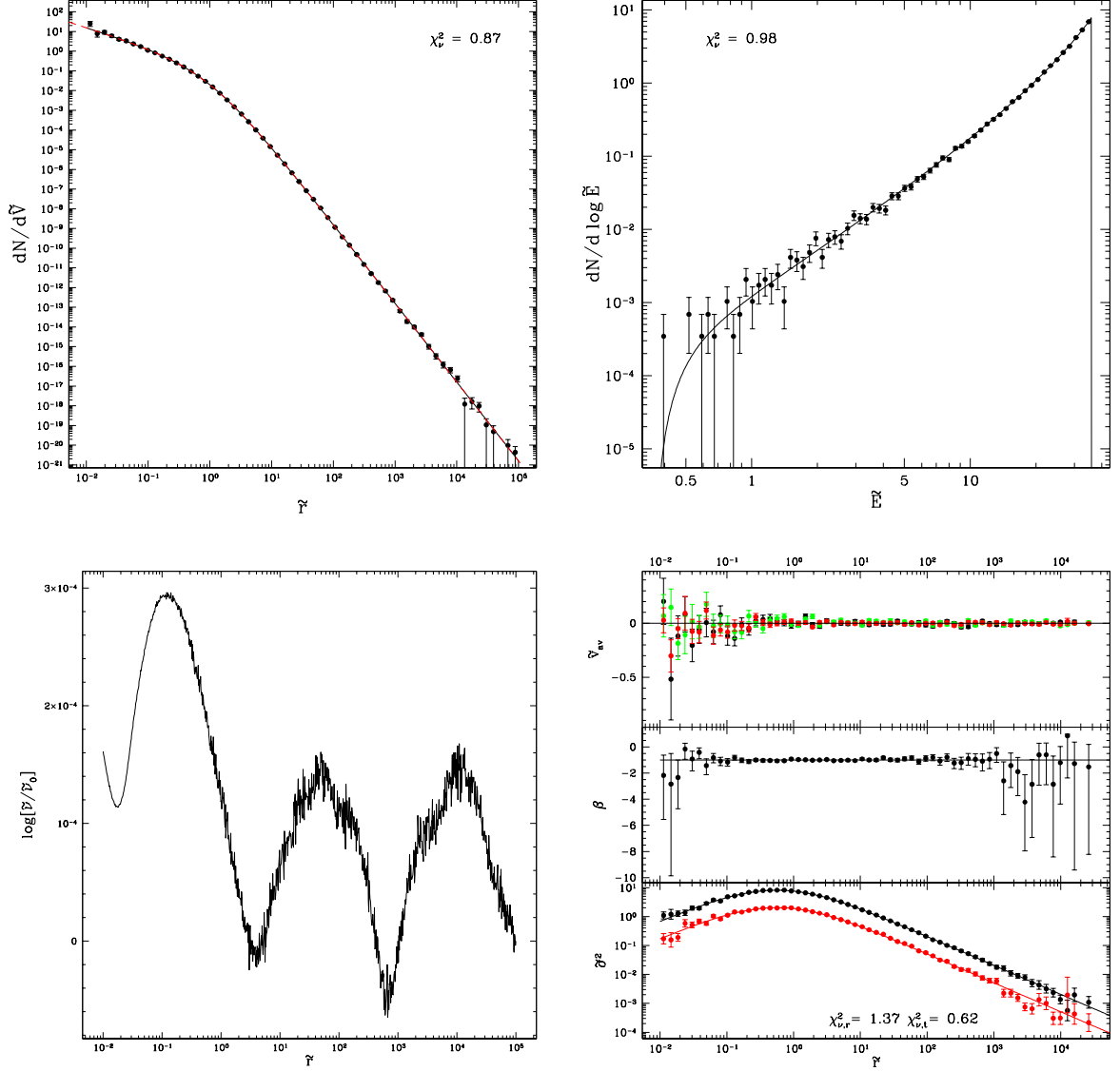


Figure 4.15: Properties of a GCS with circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $\gamma = 1$ Dehnen model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

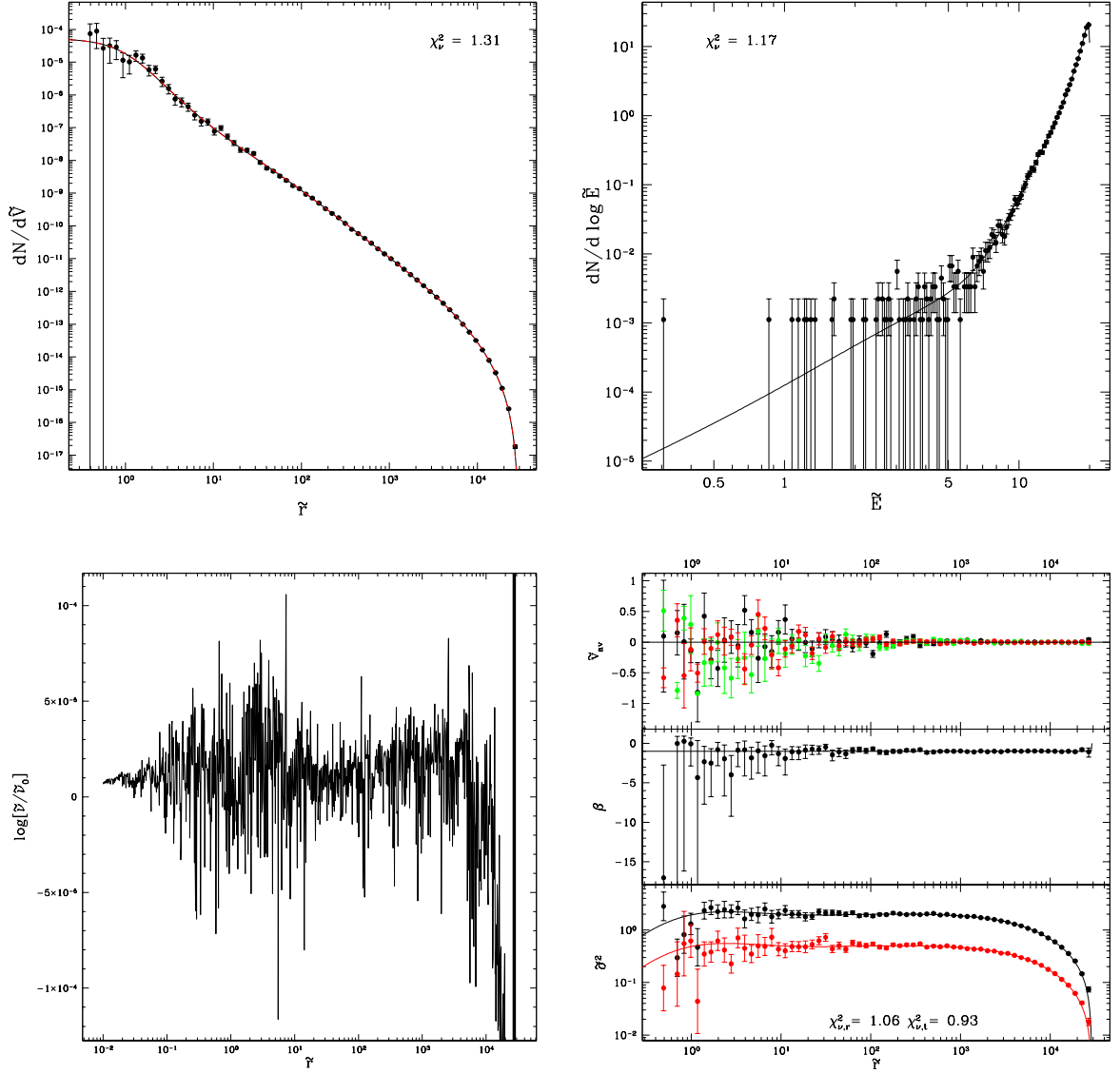


Figure 4.16: Properties of a GCS with circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $W_0 = 20$ King model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

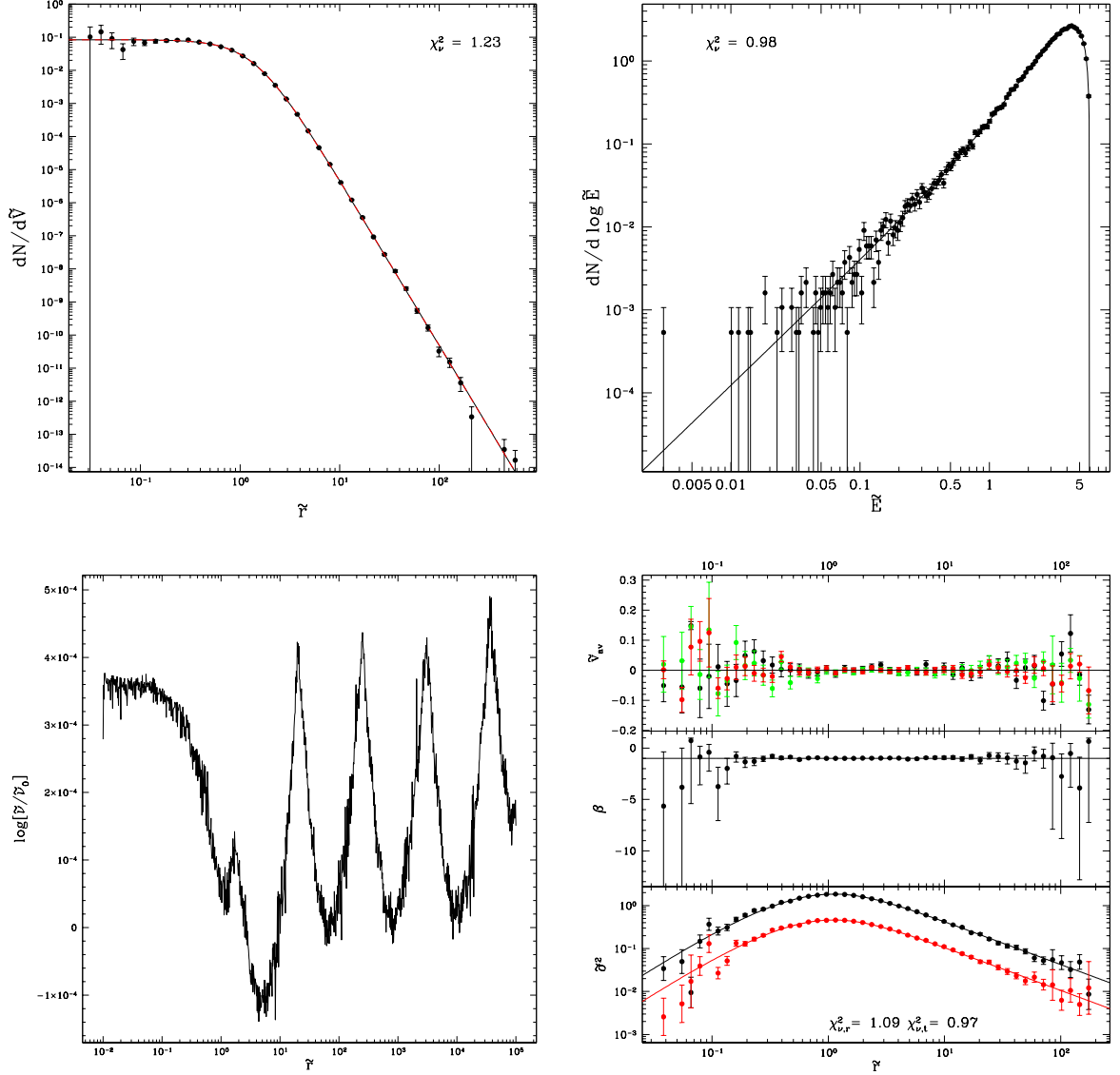


Figure 4.17: Properties of a GCS with circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $k = 2.5$ polytrope model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

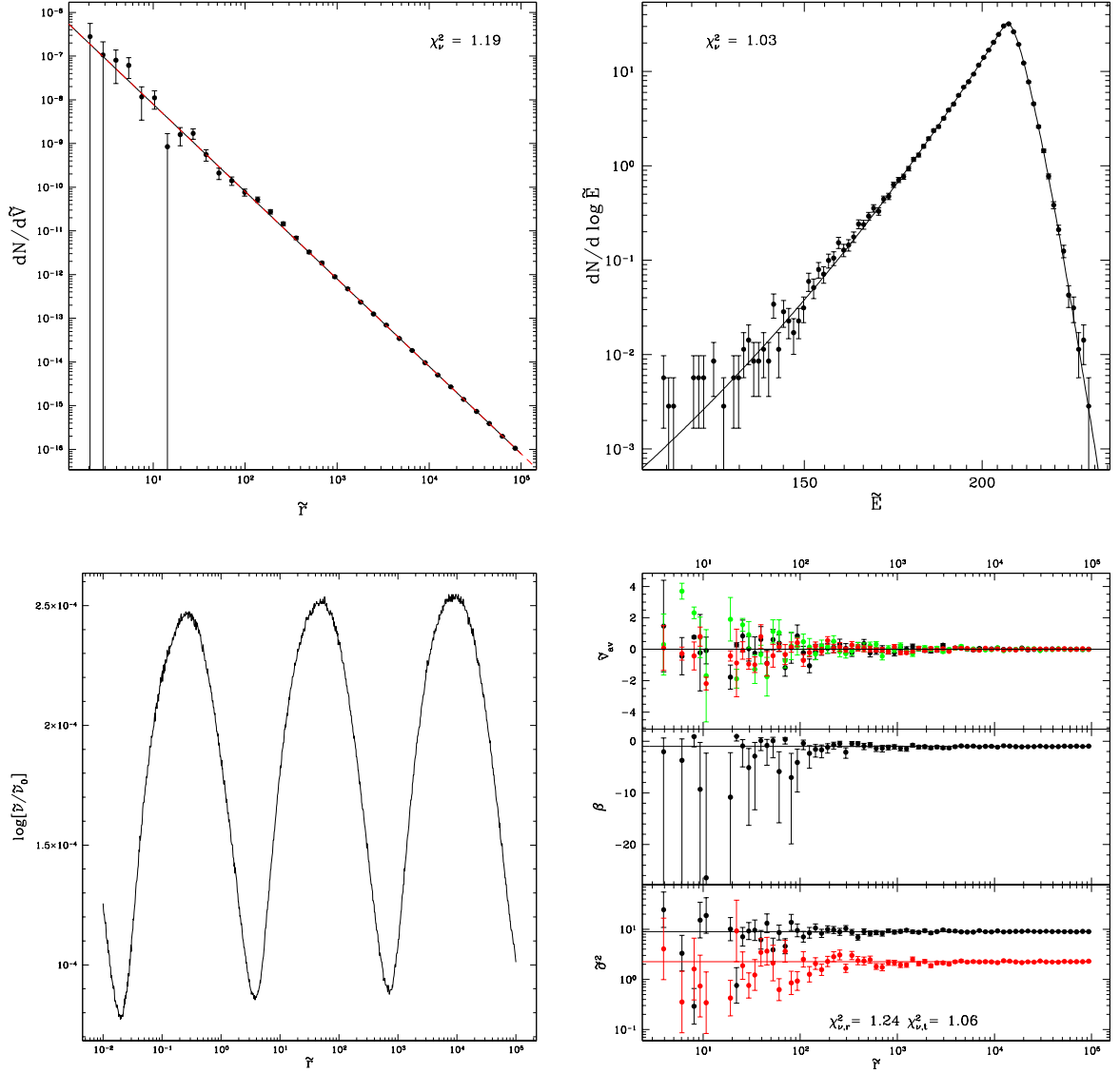


Figure 4.18: Properties of a GCS with circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$, and spatial distribution directly tracing a $k = 2$ power-law model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

Once again the sampling clearly works well, with reduced chi-squared values from the spatial, orbital energy, and velocity samplings near to unity for all the models, and typical differences of the analytic input number density from the numerically recovered one of around 0.1%. Furthermore, the average polar, azimuthal and radial velocities in tracer bins almost always sit within their error bars of zero. Additionally, the velocity anisotropy samplings clearly work as intended, displaying constant tangential anisotropy as specified with the input circularity function, closely matched by the velocity anisotropies in each tracer bin. The slight flick at the zero-density surface of the King host galaxy model in Figure 4.16 is just due to the distribution function plummeting to zero there, drowning out the relatively small difference in $\tilde{\sigma}_r^2$ and $\tilde{\sigma}_t^2$.

The next set of simulated GCSs again have constant tangential anisotropy (equation 3.57, $j(h) = h^{-2\beta_0}$ with $\beta_0 = -1$) with $\tilde{L}_0 = 0$, in the same host galaxy models as the previous set. However, rather than directly tracing their host galaxies, these GCSs follow double power-law number densities (equation 3.76, with $\gamma_{pl} = 0, \beta_{pl} = 3.5, \alpha_{pl} = 0.5$), and are displayed in Figures 4.19, 4.20 4.21 4.22.

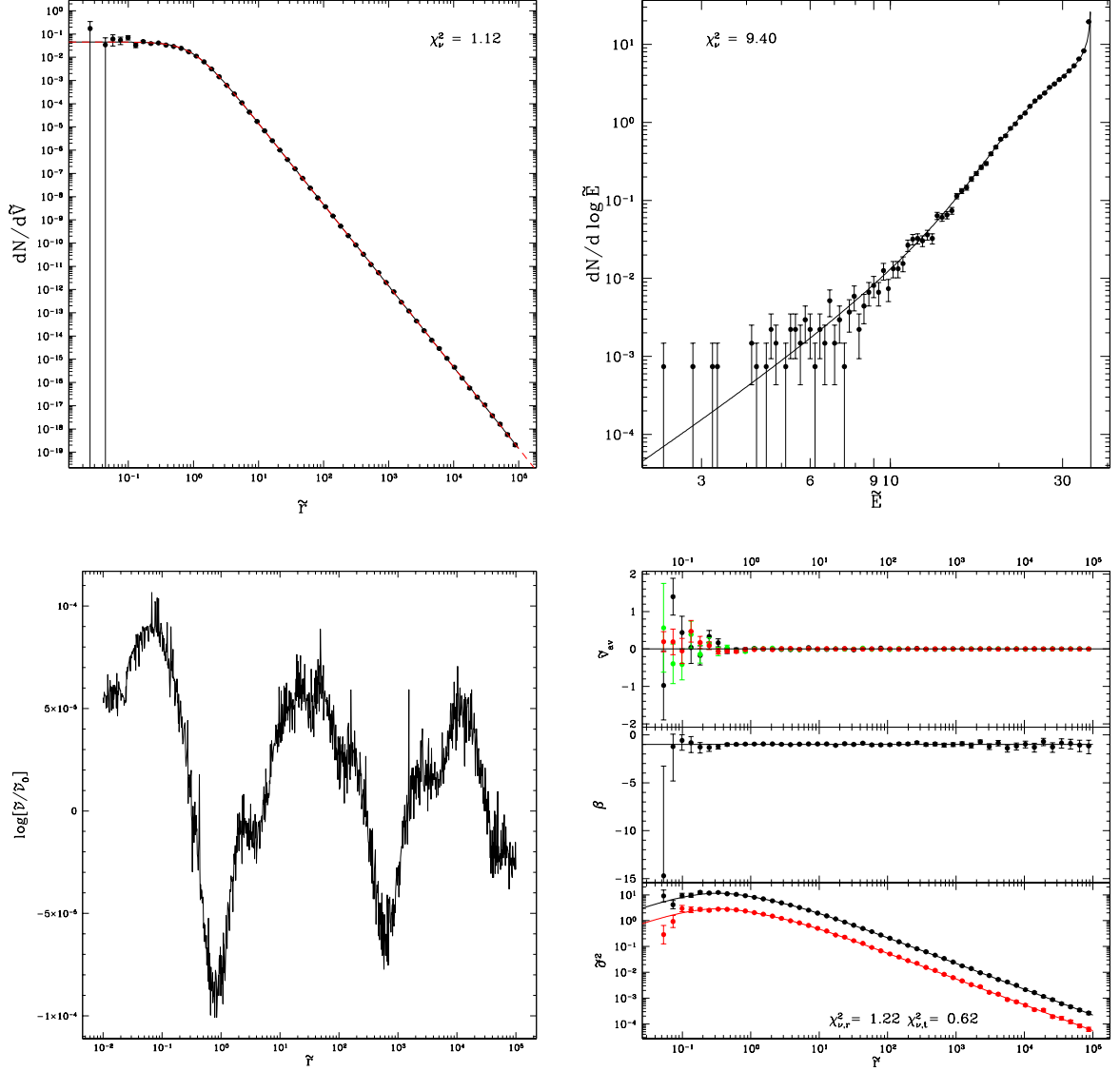


Figure 4.19: Properties of a GCS with circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$, and double power-law spatial distribution in a $\gamma = 1$ Dehnen model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

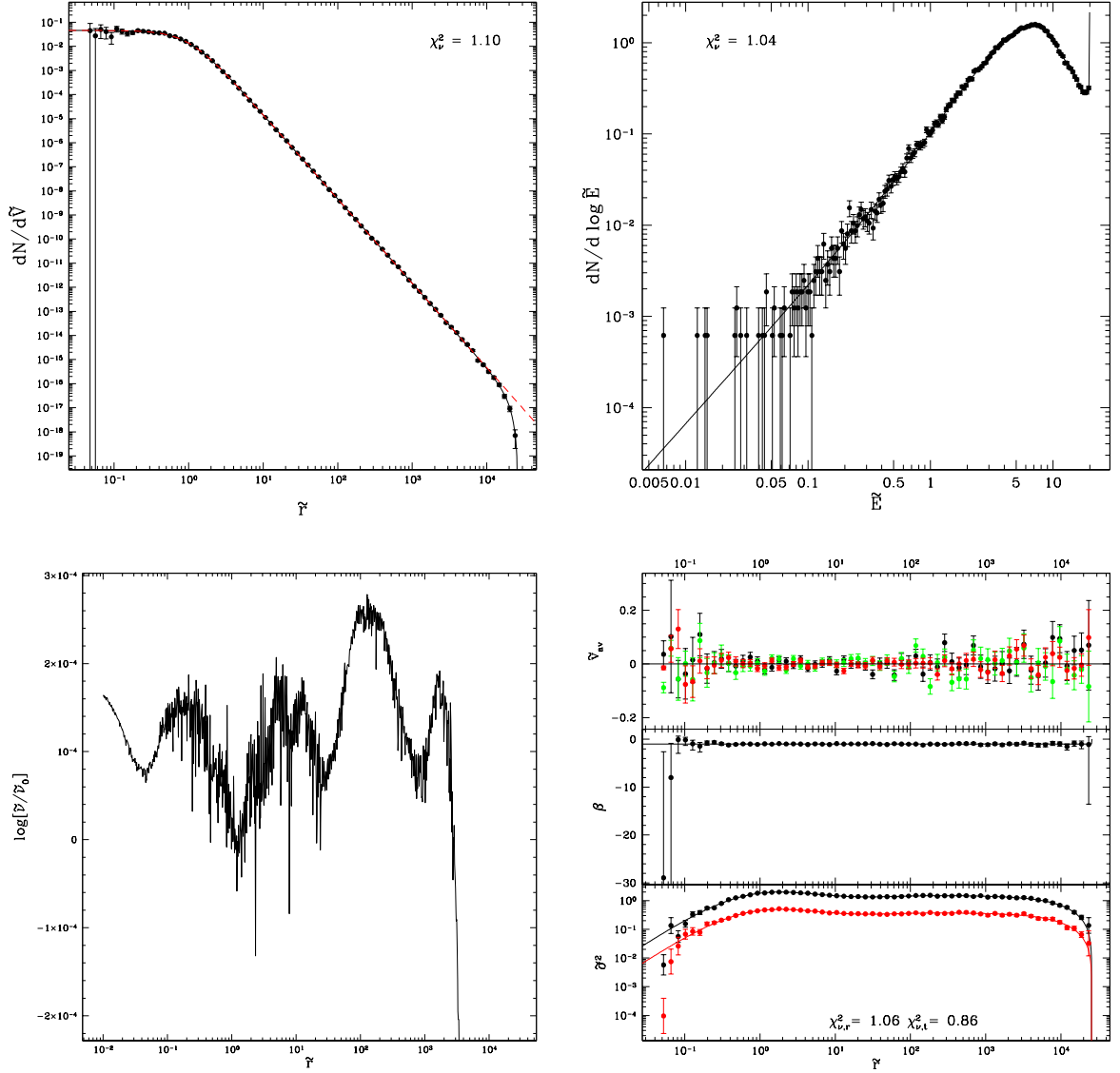


Figure 4.20: Properties of a GCS with circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$, and double power-law spatial distribution in a $W_0 = 20$ King model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

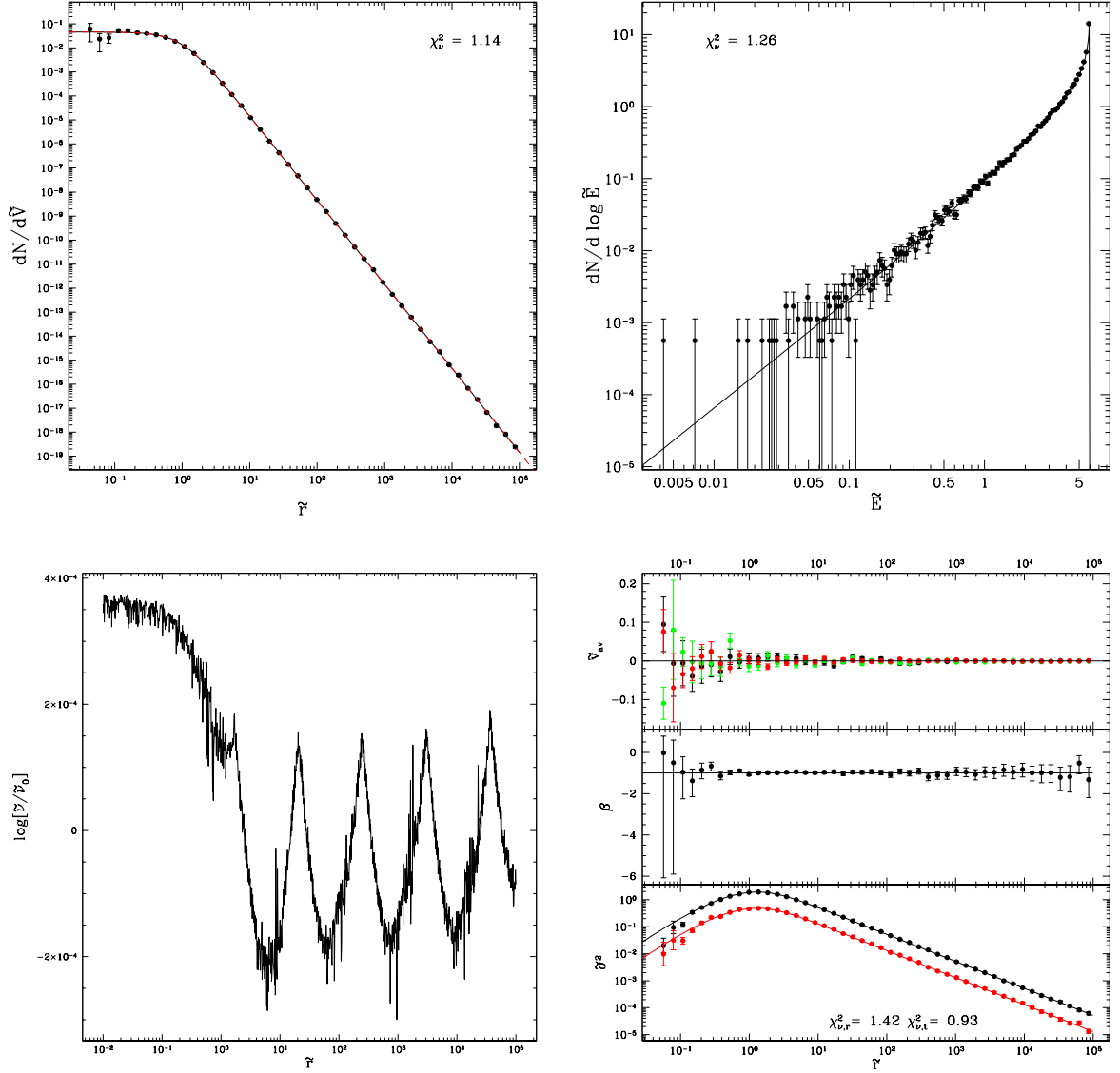


Figure 4.21: Properties of a GCS with circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$, and double power-law spatial distribution in a $k = 2.5$ polytrope model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

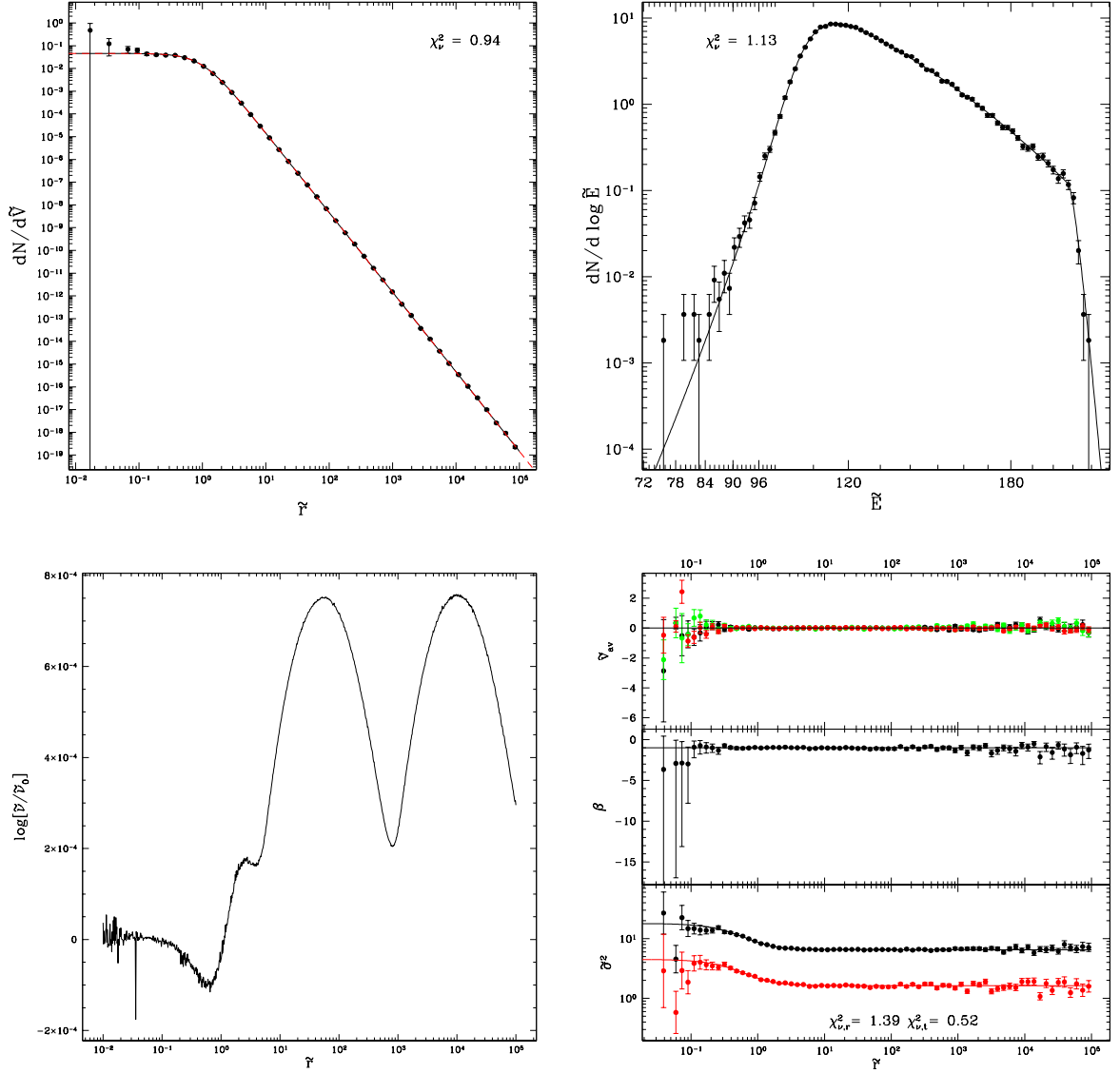


Figure 4.22: Properties of a GCS with circularity function $j(h) = Ch^{-2\beta_0}$ with $\beta_0 = -1$ and $\tilde{L}_0 = 0$, and double power-law spatial distribution in a $k = 2$ power-law model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

Clearly there are no difficulties sampling double power-law tracer spatial distributions in any of the host galaxies, with reduced chi-squared values all near to unity. Note that the GCS number density in the King host galaxy in Figure 4.20 displays a significant deviation from the double power-law profile at large radii. Physically, this is of course due to the fact that King models have finite radii, i.e. the number density must go to zero at the zero-density radius. Mathematically, this is because the upper limit on the integral over the distribution function in equation (3.50) is the value of the potential at the zero-density radius, and therefore the distribution function obtained using numerical quadrature must be zero for all energies corresponding to radii beyond the zero-density radius. In fact, the spatial distribution of all GCSs will diverge from the input tracer number density eventually (whether double power-law or tracing the host galaxy), because when numerically solving for the distribution function, the upper limit on the integral equation is taken to be the potential at some large radius, \tilde{r}_f (see section 3.4.1). For formally infinite models, this large radius is set to be $\tilde{r}_f = 10^{10}$, and is therefore well beyond the edge of the plots displayed here. Aside from this divergence in the King host galaxy, the differences between input number density and numerically recovered number density are all less than 0.2%. The differential energy distribution of the GCS in the $\gamma = 1$ Dehnen host galaxy in Figure 4.19 displays a reduced chi-squared of 9.27. Similarly to the $k = 2.5$ power-law in Figure 4.14, this is completely due to the rightmost tracer bin and the steeply increasing differential energy distribution. Again, this could be accounted for with more finely spaced tracer bins, but again doing so dramatically increases the noise at lower energies. Aside from this one bin, the sampled tracer orbital energies generally match the differential energy distribution very well, and the differential energy distributions of the rest of the GCSs in this set are all near unity. Moreover, the velocity dispersions of both the sampled radial and tangential velocities match the curves very well, with reduced chi-squared values comfortably near to unity, and unsurprisingly combine to produce bins with velocity anisotropies that match the curve well. Additionally, the average radial, azimuthal and polar velocities all cluster around zero, indicating no problems with net rotation or flux.

The next set of plots are designed to test the sampling in simulated GCSs with circularity function $j(h) = \exp(-ah)$, with $a = 4$ and $\tilde{L}_0 = 0$. This is so that the resulting anisotropy profile is something more complicated than constant anisotropy, while keeping the other properties the same as the previous set (i.e. in the same host galaxies and still with double power-law number density profiles). Since $\frac{dj}{dh} < 0$ is expected to give a radially biased velocity profile, the circularity function $j(h) = \exp(-ah)$ would be expected to result in an anisotropy profile that increases with galactocentric distance. The properties of these simulated GCSs are displayed in Figures 4.23, 4.24, 4.25 and 4.26.

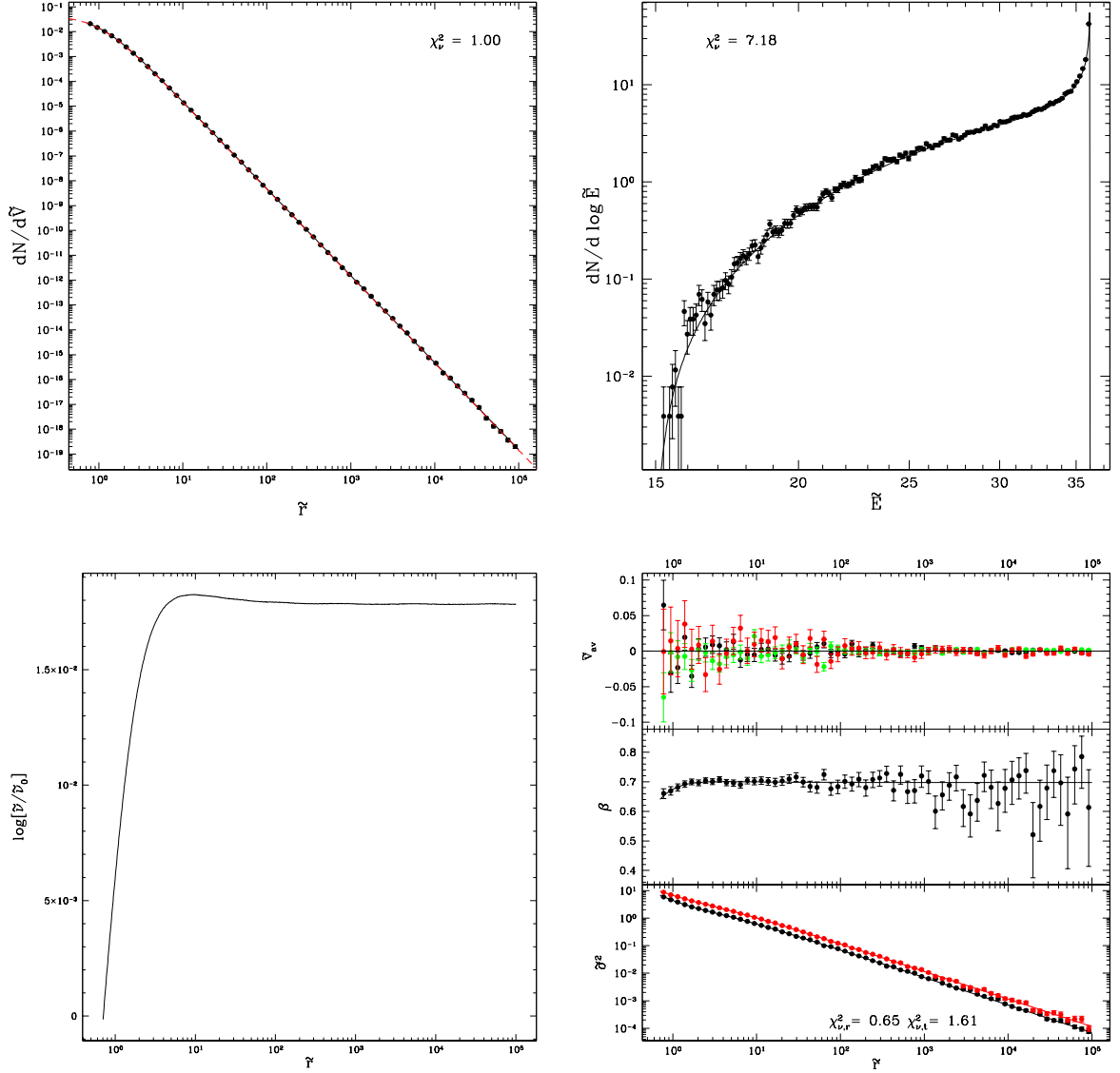


Figure 4.23: Properties of a GCS with circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 0$, and double power-law spatial distribution in a $\gamma = 1$ Dehnen model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

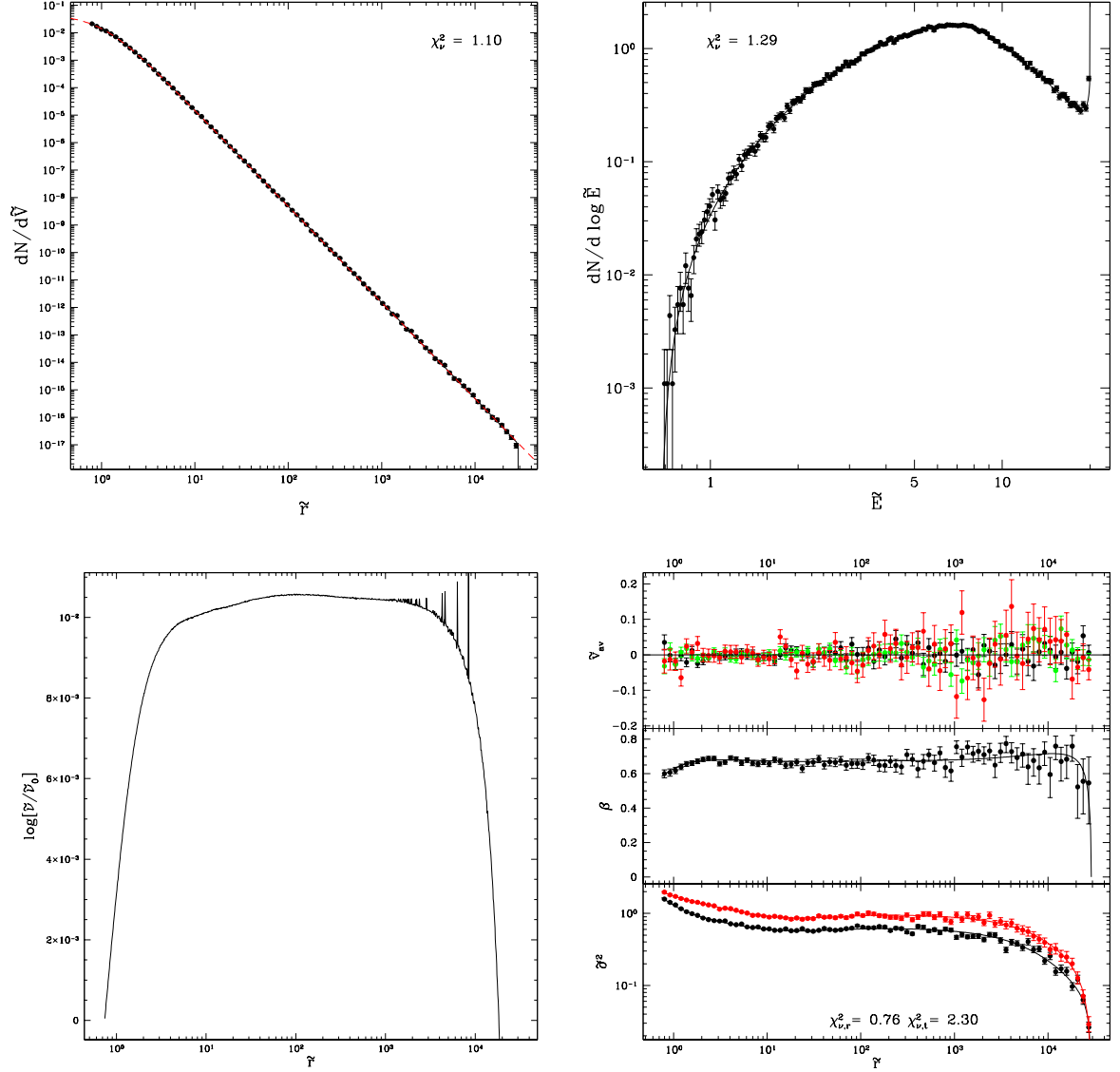


Figure 4.24: Properties of a GCS with circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 0$, and double power-law spatial distribution in a $W_0 = 20$ King model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

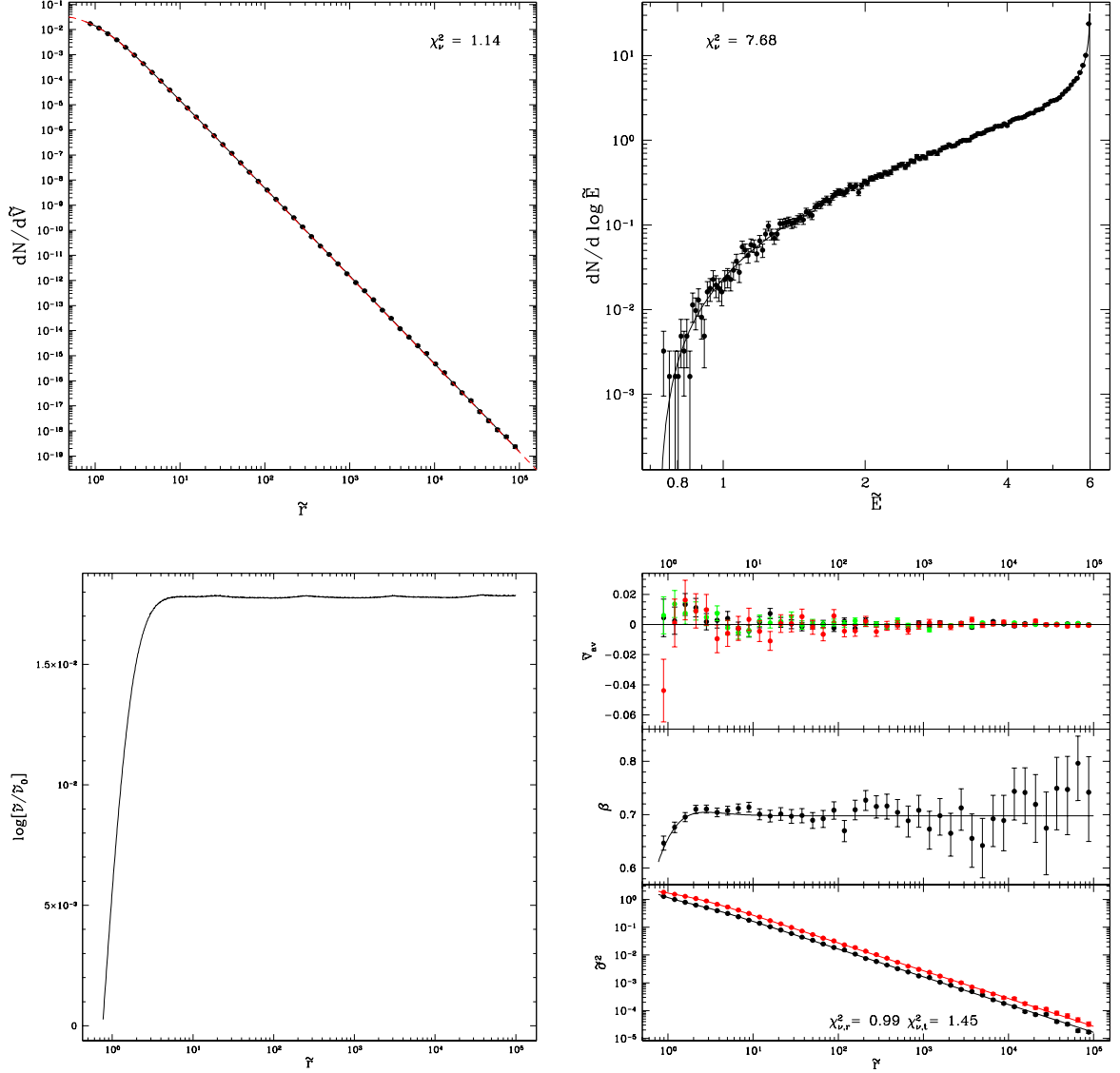


Figure 4.25: Properties of a GCS with circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 0$, and double power-law spatial distribution in a $k = 2.5$ polytrope model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

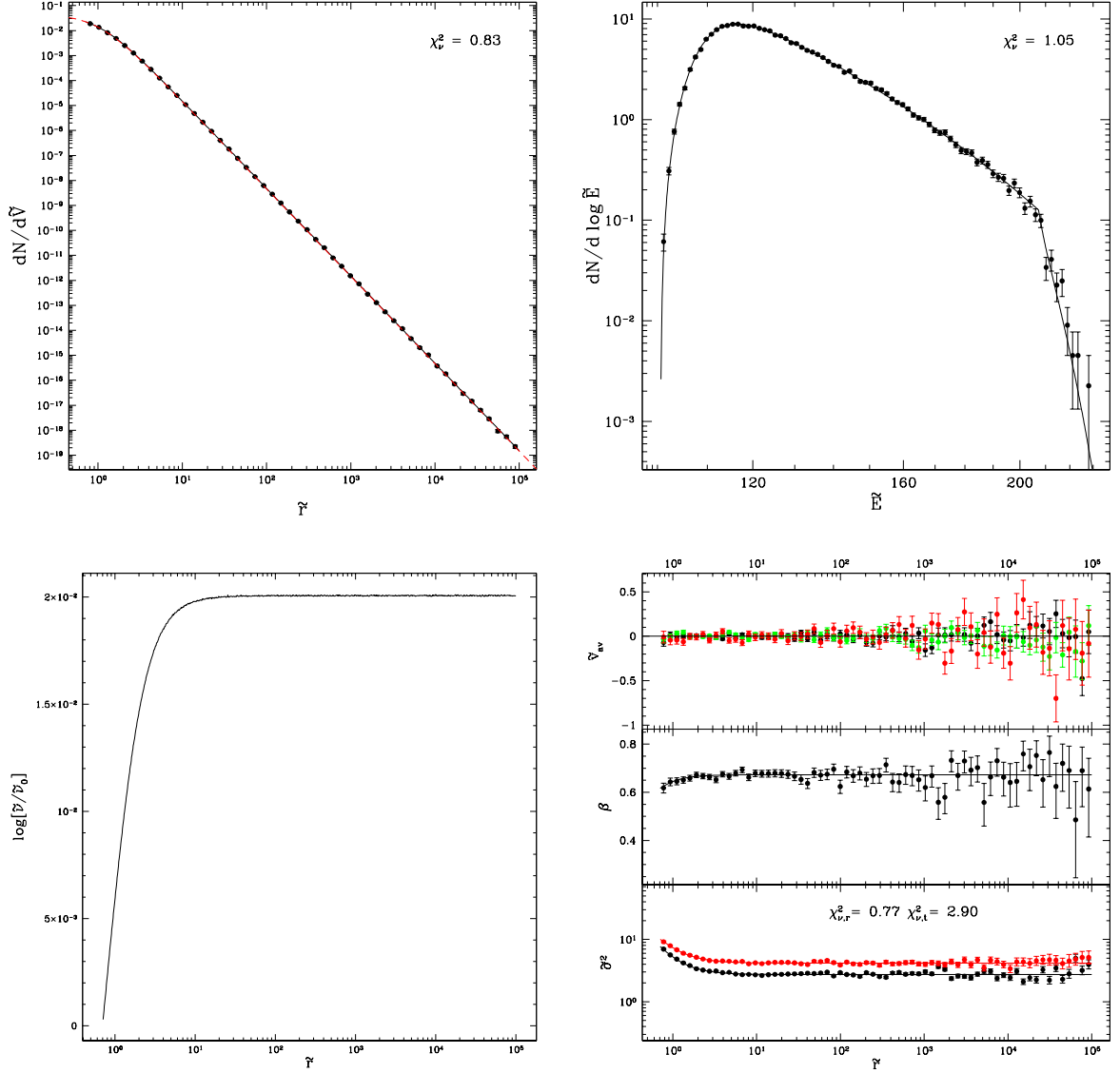


Figure 4.26: Properties of a GCS with circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 0$, and double power-law spatial distribution in a $k = 2$ power-law model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

Once again the number density sampling is very satisfactory, with all reduced chi-squared values around unity. However, clearly the more complicated circularity function has made solving the integral equation more difficult (cf. equation 3.50), as the number density obtained by integrating over the numerically solved distribution function typically differs from the input number density by about 4%, as opposed to a typical difference of less than 1% for the simpler models. In any event, this is not a serious concern, as there will be other bigger sources of error when applying simulated GCSs to projected observed GCSs, and moreover the ratio of numerically recovered to input number density appears to be constant at large radii in each of the host galaxies (except of course the King model host galaxy). Once again, the high values of reduced chi-squared in Figures 4.23 and 4.25 is a binning issue rather than reflecting a problem with the sampling, caused by the steeply increasing differential energy distributions at high energies. The average polar, azimuthal and radial velocities clustering around zero at all radii indicates that the more complicated circularity function has not caused any obvious bias in the velocity samplings. Furthermore, the velocity anisotropies are behaving as expected based on having a circularity function that decreases with h , i.e. all the simulated GCSs display increasing radial velocity anisotropy, tending to a constant of $\beta \simeq 0.7$ outside of the core (i.e. $\tilde{r} \gtrsim 1$) in each host galaxy. The tangential velocity dispersions display slightly elevated reduced chi-squared values, which might be expected given that the circularity function is depleting tangential motions quite strongly. In any event, inspection of the plots reveals that the velocity dispersions of the sampled tangential velocities are in fact following the curve quite closely, though with slightly more noise than their radial counterparts, and so these slightly elevated reduced chi-squared values could just be due to the binning.

The next set of plots display simulated GCSs with double power-law number densities in the same host galaxies, and with circularity function $j(h) = \exp(-ah)$ with $a = 4$, but this time with $\tilde{L}_0 = 100$. Having custom number density profiles and anisotropic velocity distributions and with a non-zero \tilde{L}_0 means that these GCSs have all features enabled and thus represent maximum complexity for the sampling. The results are displayed in Figures 4.27, 4.28, 4.29, 4.30.

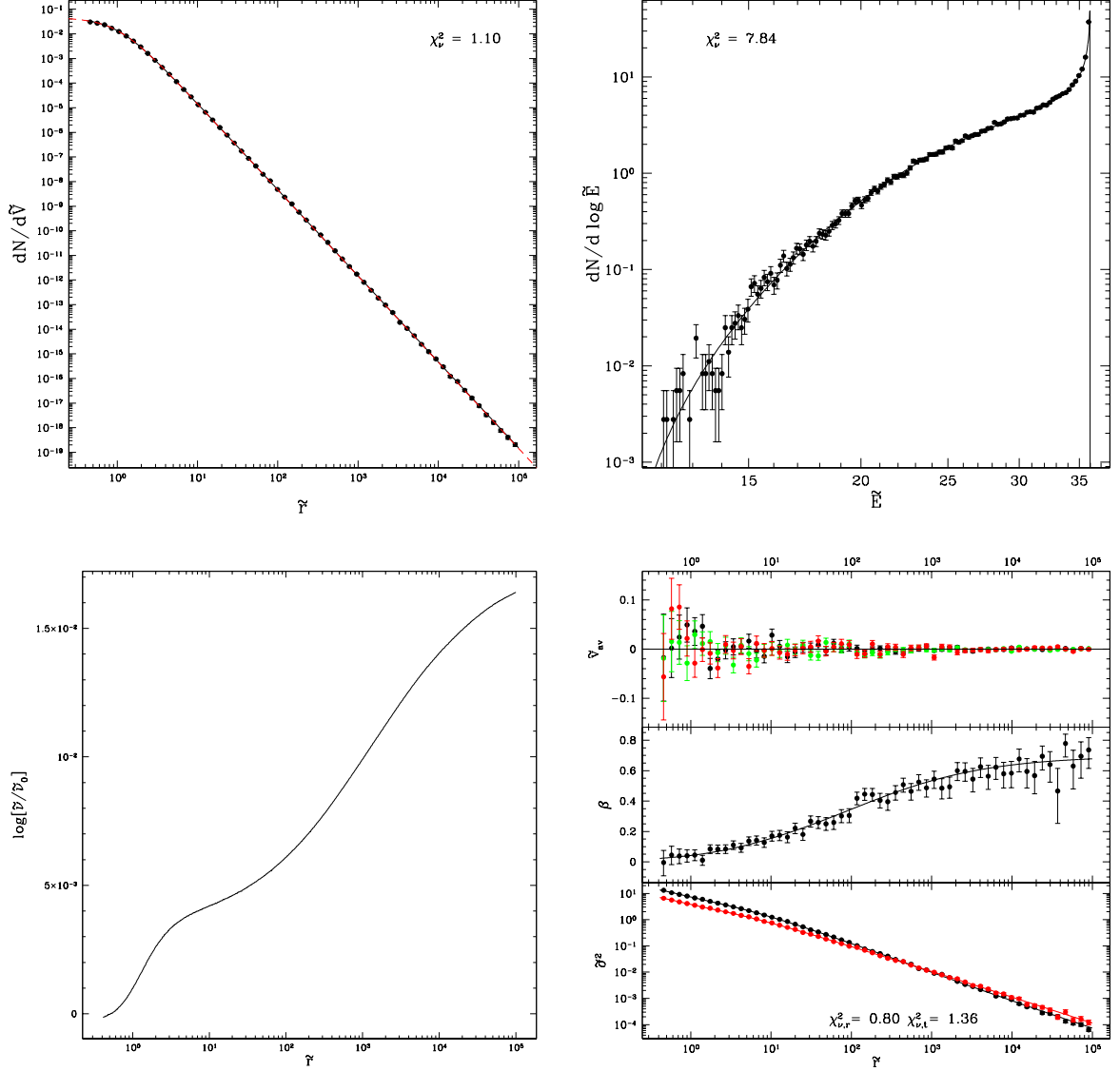


Figure 4.27: Properties of a GCS with circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 100$, and double power-law spatial distribution in a $\gamma = 1$ Dehnen model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

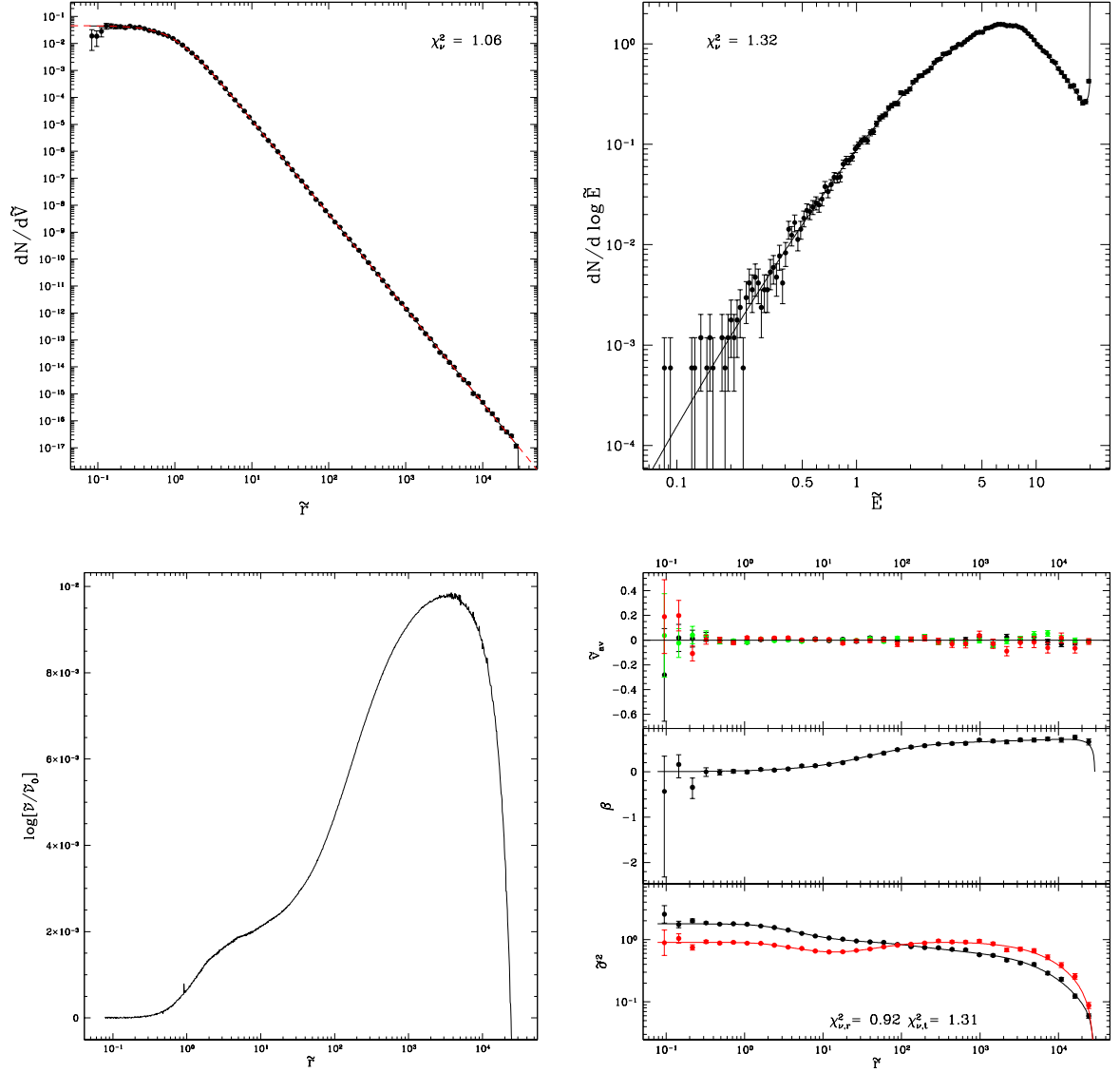


Figure 4.28: Properties of a GCS with circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 100$, and double power-law spatial distribution in a $W_0 = 20$ King model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

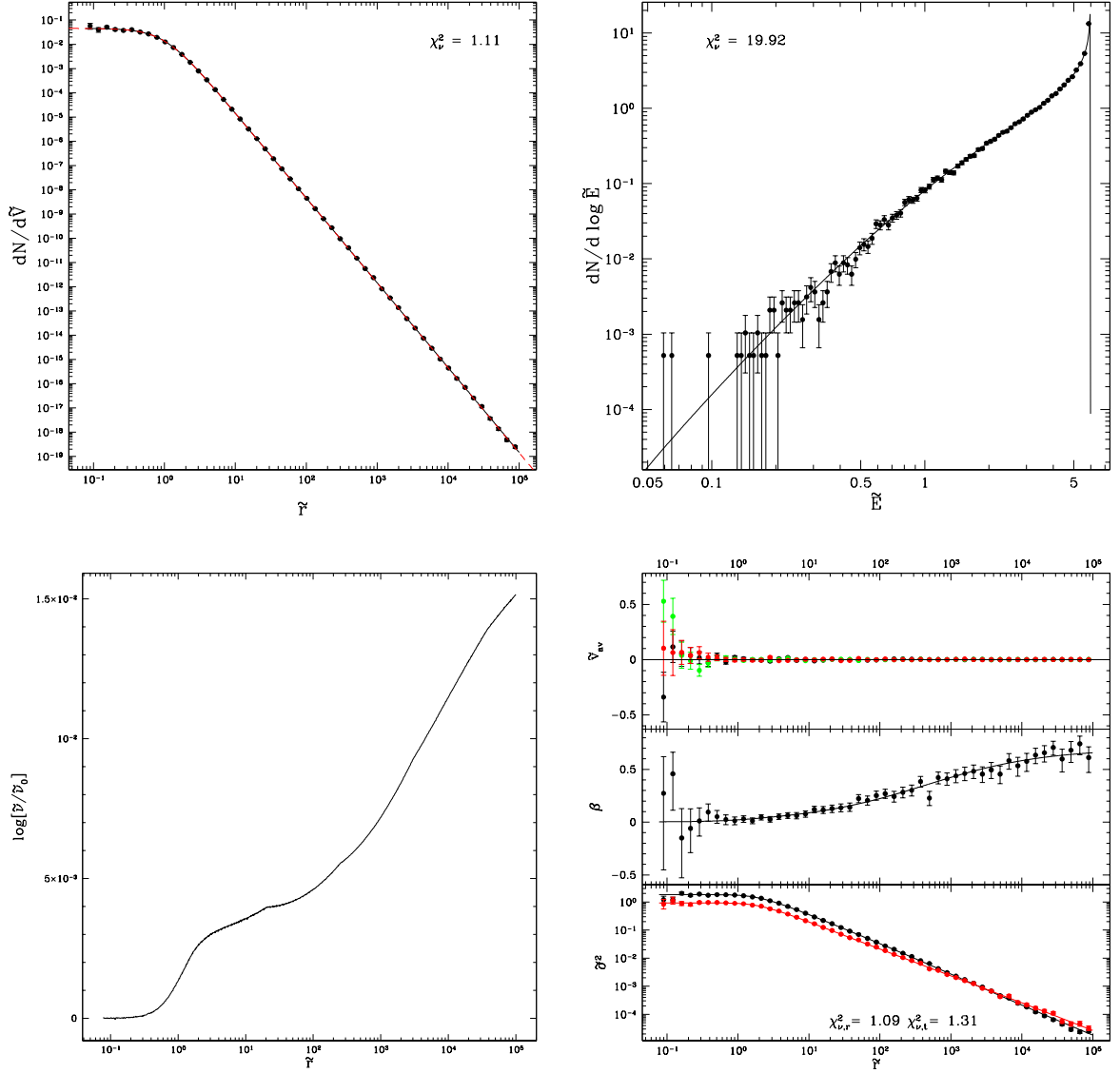


Figure 4.29: Properties of a GCS with circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 100$, and double power-law spatial distribution in a $k = 2.5$ polytrope model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

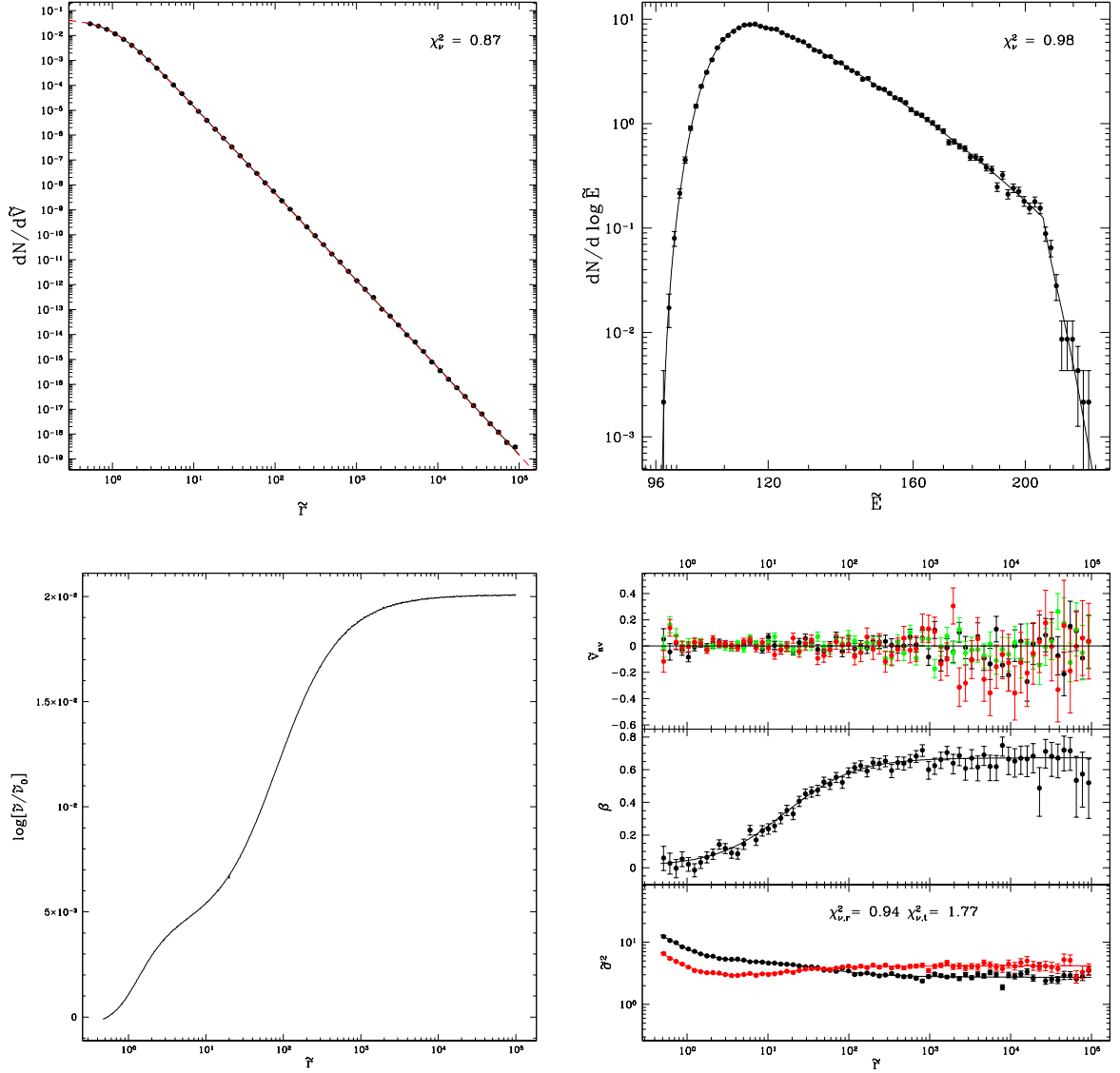


Figure 4.30: Properties of a GCS with circularity function $j(h) = \exp(-ah)$ with $a = 4$ and $\tilde{L}_0 = 100$, and double power-law spatial distribution in a $k = 2$ power-law model host galaxy. In the top left, the black curve is the GCS number density corresponding to the distribution function and the red dashed curve is the input number density. The reduced chi-squared indicates how well the binned sampled tracers (black points) match the black curve. The bottom left plot depicts the ratio of numerical number density to input number density (black and red dashed curves above). The top right plot displays the differential energy distribution and bins of sampled tracer orbital energies, along with the reduced chi-squared indicating how well they match. The bottom right plot is split into three panels. In descending order these are: the averages of each component of velocity; the velocity anisotropies; the velocity dispersions. Red points correspond to radial velocities, green to polar, and black to azimuthal. In the bottom panel, the curves and points denote numerical and sampled σ_r^2 (red) and σ_t^2 (black). Additionally the reduced chi-squared values indicating how well the sampled points match the numerical curve for both radial and tangential velocity dispersions are given.

And again for the final set of test plots, the number density sampling works very satisfactorily with all reduced chi-squared values near unity. The difference between input and numerically recovered number densities is again at most about 4%, however this time it is less clear whether it will tend to a constant at large radii. The bottom left plot in Figures 4.27 and 4.29 both show the ratio of numerically recovered to input number density steadily increasing with galactocentric radius. On the other hand, given that the behaviour of this ratio in each of the host galaxies has hitherto been broadly similar (barring the rapid divergence at high radii in King model host galaxies), it seems plausible that the curves in Figures 4.27 and 4.29 will flatten out and resemble the curve in Figure 4.30. Nevertheless, even if this is not the case and the numerically recovered number density continues to diverge from the input number density, it is unlikely that any adversely affected data would actually be needed, as the difference is only about 4% at $\tilde{r} = 10^5$, and given the roughly constant increase with $\log \tilde{r}$ is likely only to become significant at radii larger than a GCS can realistically be expected to extend out to.

As might be expected given that these simulated GCSs are very similar to the previous set, the differential energy distributions are very similar, including the problematic sharp peaks at high energies for the Dehnen and polytrope host galaxies in Figures 4.27 and 4.29. Additionally, the average radial, polar and azimuthal velocities again cluster around zero with no obvious systematics, indicating that the sampling works without introducing any bias in even the most complicated simulated GCSs. Furthermore, the velocity dispersions of the sampled radial and tangential velocities match the numerical curves very well, with reduced chi-squared values comfortably near unity. This could indicate that the slightly elevated reduced chi-squared values in the previous set of plots was indeed due to the radial anisotropy, since in this case the increase towards radial anisotropy is much more gradual. Given that \tilde{L}_0 can be thought of as an anisotropy radius multiplied by a characteristic system speed, increasing \tilde{L}_0 might be expected to force isotropy in the core. This appears to be the case, with the velocity anisotropies in each host galaxy remaining isotropic out to different radii (as might be expected since different gravitational potentials will lead to different charac-

teristic speeds), before becoming radially anisotropic and tending to roughly $\beta \simeq 0.7$ at large radii.

In summary, these demonstrations clearly show that the initial GCS set-up code outlined in section 3.7 works very well, accurately sampling tracer GCs from a variety of GCS number density profiles and circularity functions, and with non-zero \tilde{L}_0 , in a variety of host galaxies, with no obvious systematics or biases.

4.2 Velocity Anisotropy and the Circularity Function

The following plots no longer contain any sampled tracers, instead only depicting numerically solved anisotropy curves. This is to explore the anisotropy profiles generated with the different circularity functions listed in section 3.5.1 (except the circularity function for constant anisotropy, since it was analytically derived and thus well understood). Based on the sets of plots displayed previously, the behaviour of anisotropy profiles of GCSs with the same properties in different host galaxies appear to be broadly similar. Therefore, only anisotropy profiles of simulated GCSs in a $\gamma = 1$ Dehnen model host galaxy are shown. Furthermore, in order to clearly ascertain the impact of changes to the circularity function on the anisotropy profile, each of the simulated GCSs follow the same spatial profile; namely directly tracing the host galaxy.

Each panel of Figure 4.31 displays the anisotropy profiles of GCSs with different circularity functions and $\tilde{L}_0 = 0$, that directly trace a $\gamma = 1$ Dehnen host galaxy. The multiple curves in each panel correspond to different values of a in the circularity function; these are: $a = 2$ (solid), $a = 3$ (dotted), and $a = 4$ (dashed). The circularity function used for the anisotropy profiles in the top panel was that given by equation (3.58), $j(h) = \exp(-ah)$. Since we have $\frac{dj}{dh} = -a \exp(-ah)$, and a circularity function that decreases with h is expected to correspond to an anisotropy profile that is radially biased, the expectation is that this circularity function will lead to radially anisotropic GCSs. Furthermore, increasing a at fixed h makes $\frac{dj}{dh}$ more negative; i.e. the circularity

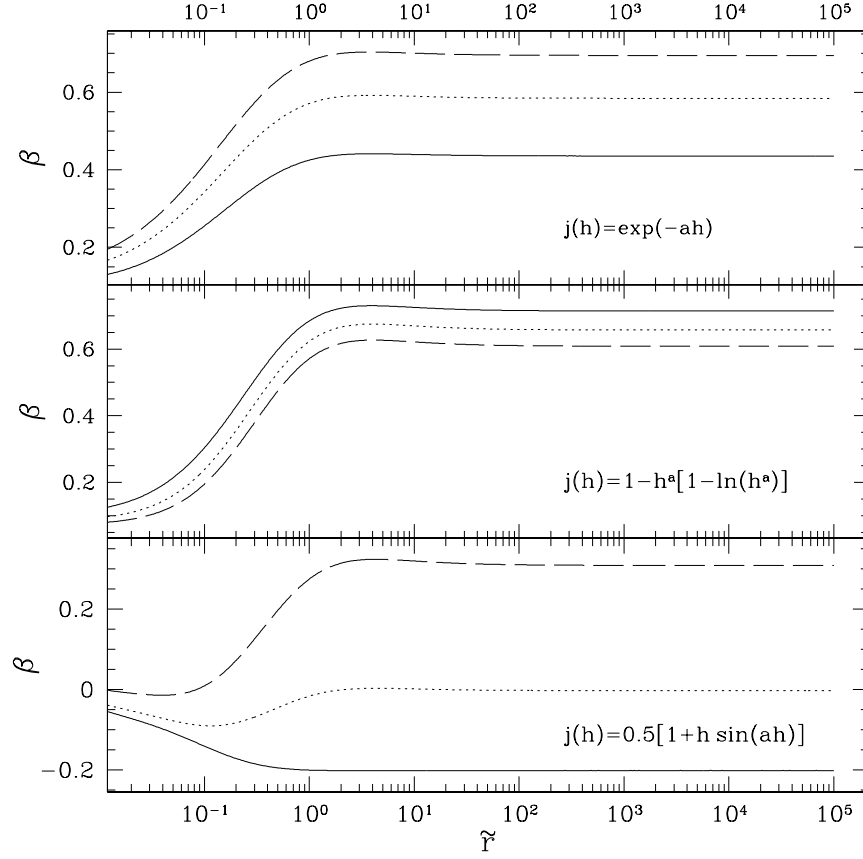


Figure 4.31: Anisotropy profiles of GCSs directly tracing a $\gamma = 1$ Dehnen model host galaxy and $L_0 = 0$, with circularity functions $j(h) = \exp(-ah)$, $j(h) = 1 - h^a[1 - \ln(h^a)]$, and $j(h) = 0.5[1 + h \sin(ah)]$ in descending order. Each panel displays the anisotropy profiles generated with $a = 2, 3, 4$ (solid, dotted, dashed respectively).

function decreases more strongly with h . Consequently, we might expect that increasing a will make the corresponding anisotropy profile more radially biased, and indeed this is what is seen. The anisotropy profiles in the middle panel were generated with a circularity function given by equation (3.60), $j(h) = 1 - h^a[1 - \ln(h^a)]$. In this case, $\frac{dj}{dh} = a^2 h^{a-1} \ln(h)$, which is always negative since $0 \leq h \leq 1$. Therefore, the expectation is that this circularity function will also produce radially biased anisotropy profiles. Increasing a at fixed h has the effect of making $\frac{dj}{dh}$ less negative; i.e. the circularity

function decreases with h more weakly. Thus, we might expect that increasing a will lead to anisotropy profiles that display less radial bias, and indeed this is reflected in the curves. The bottom panel displays anisotropy profiles corresponding to a circularity function given by equation 3.61, $j(h) = 0.5[1 + h \sin(ah)]$. In this case, we have $\frac{dj}{dh} = 0.5[\sin(ah) + ah \cos(ah)]$, which has roots at $h = -2.02876/a$, 0 , and $2.02876/a$, and therefore $\frac{dj}{dh} > 0$ for $h < -2.02876/a$ and $0 < h < 2.02876/a$. Therefore, for $a = 2$ (solid curve), we have that $\frac{dj}{dh} > 0$ for $0 \leq h \leq 1$, and consequently would expect tangential anisotropy, which is exactly what is seen. However, for $a > 2.02876$, such as the dotted and dashed curves ($a = 3$ and $a = 4$ respectively), $\frac{dj}{dh}$ starts positive but turns negative within the interval $0 \leq h \leq 1$, and therefore the behaviour of the velocity anisotropy is much less obvious. In general, the average gradient of the circularity function in the interval $0 \leq h \leq 1$ is given by $\frac{\Delta j}{\Delta h} = [j(1) - j(0)]/[1 - 0]$, which in this case is $\frac{\Delta j}{\Delta h} = 0.5 \sin(a)$. For $a = 2, 3, 4$, this corresponds to an average gradient of $0.45, 0.07, -0.38$. This seems to reflect the asymptotic behaviour of the velocity anisotropy rather well; the $a = 2$ curve (solid) displays clear tangential anisotropy, the $a = 3$ curve (dotted) displays very slight tangential anisotropy, and the $a = 4$ curve (dashed) displays clear radial anisotropy.

As the circularity function given by equation (3.59), $j(h) = \exp[-(h-a)^2/2b^2]$, is a function of two parameters (a and b), the corresponding anisotropy profiles are given in Figure 4.32, with the same 3 values of a ($0, 0.5, 1$, corresponding to solid, dotted, and dashed respectively) with a different b ($0.5, 1, 2$, in descending order) in each panel. Since a represents the mean value of the Gaussian, for $a = 0$ (solid curves), the circularity function will monotonically decrease with h , and thus we might intuitively expect radial anisotropy as a result. Conversely, for $a = 1$ (dashed curves), the circularity function will monotonically increase with h , and thus tangential anisotropy is expected as a result. However, for $a = 0.5$ (dotted curves), the circularity function in the interval $0 \leq h \leq 1$ is symmetrical, and consequently we might naïvely expect isotropy in the resulting anisotropy profiles. Clearly this is not the case, with the dotted curves in every panel displaying radial anisotropy (albeit fairly weak). This is not surprising since obviously in reality the dependence of the anisotropy profile on the circularity

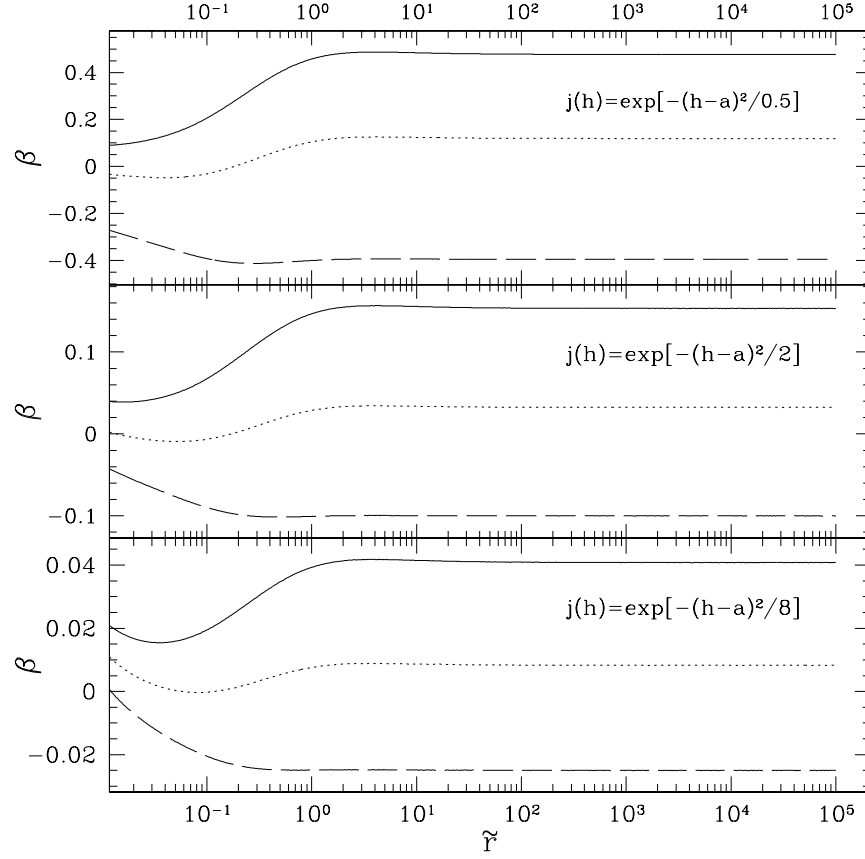


Figure 4.32: Anisotropy profiles of GCSs directly tracing a $\gamma = 1$ Dehnen model host galaxy, with circularity function $j(h) = \exp[-(a - h)^2/2b^2]$. In descending order each panel has $b = 0.5, 1, 2$, and in every panel are displayed the anisotropy profiles generated with $a = 0, 0.5, 1$ (solid, dotted, dashed respectively).

function will be more complicated than just the average gradient. For example, since $j(h)$ is integrated over the range $1 \leq h \leq h_{max}$ with $h_{max} = \sqrt{2(\tilde{E} - \tilde{\Phi})\tilde{r}/[\tilde{L}_c(\tilde{E}) + \tilde{L}_0]}$, the asymptotic behaviour will likely depend on properties of the host galaxy such as $\tilde{\Phi}$ and $\tilde{L}_c(\tilde{E})$. However, the average gradient of the circularity function does seem to broadly reflect the behaviour of the asymptotic anisotropy, and is therefore a useful tool that will continue to be used.

In the case of a circularity function given by $j(h) = \exp[-(a - h)^2/2b^2]$, the

average gradient is given by $\frac{\Delta j}{\Delta h} = \exp[-(1-a)^2/2b^2] - \exp[-a^2/2b^2]$, which as expected is negative for $a < 0.5$ and positive for $a > 0.5$. Note that replacing a with $1-a'$ makes this $\frac{\Delta j}{\Delta h}_{a=1-a'} = \exp[-a'^2/2b^2] - \exp[-(1-a')^2/2b^2] = -\frac{\Delta j}{\Delta h}$, i.e. the dependence of the average gradient on b is equal in magnitude but opposite in sign, with the point of symmetry at $a = 0.5$. The precise dependence of $\frac{\Delta j}{\Delta h}$ on b depends on the exact value of a , with a stationary point in $\frac{\Delta j}{\Delta h}$ as a function of b at:

$$b_0(a) = b_0(1-a) = \sqrt{\frac{1-2a}{4 \ln\left(\frac{1-a}{a}\right)}} \quad (4.5)$$

Consequently, the absolute value of $\frac{\Delta j}{\Delta h}$ increases with b up to b_0 , and then decreases asymptotically towards zero for larger values of b . Note that for both $a = 0$ and $a = 1$, $b_0 = 0$ and so $\frac{\Delta j}{\Delta h}$ monotonically increases/decreases with b , as expected. In terms of anisotropy profiles, this means that increasing b up to b_0 should result in stronger radial/tangential anisotropy, and increasing b past b_0 should result in weaker radial/tangential anisotropy, depending on whether $a < 0.5$ or $a > 0.5$, respectively. In Figure 4.32, $a = 0, 0.5, 1$, and so there is no complication due to b_0 (since both $a = 0$ and $a = 1$ result in $b_0 = 0$, and $a = 0.5$ corresponds to $\frac{\Delta j}{\Delta h} = 0$ for all b). Therefore, moving a through the sequence $0 \rightarrow 0.5 \rightarrow 1$ (solid, dotted, dashed curves respectively) will result in anisotropy profiles that are strongly radial \rightarrow roughly isotropic \rightarrow strongly tangential, and as b increases down the panels, the velocity anisotropy (be it radial or tangential) will be weaker, which is exactly what the curves reflect.

As with the previous anisotropy profile plots, Figure 4.33 displays the anisotropy profiles of a simulated GCS that directly traces a $\gamma = 1$ Dehnen model host galaxy, but this time the anisotropy profiles in all three panels were produced with the circularity function $j(h) = 0.5[1 + h \sin(ah)]$, but with $a = -2, -3, -4$ (solid, dotted and dashed respectively) in every panel, and $\tilde{L}_0 = 0, 10, 50$ in descending order. The reasons for this are twofold; firstly, to demonstrate the effects of increasing \tilde{L}_0 with circularity functions producing relatively complicated anisotropy profiles, and secondly to demonstrate the effect that mirroring the circularity function has on the resulting anisotropy profile. The bottom panel of Figure 4.31 and the top panel of Figure 4.33 both have identical

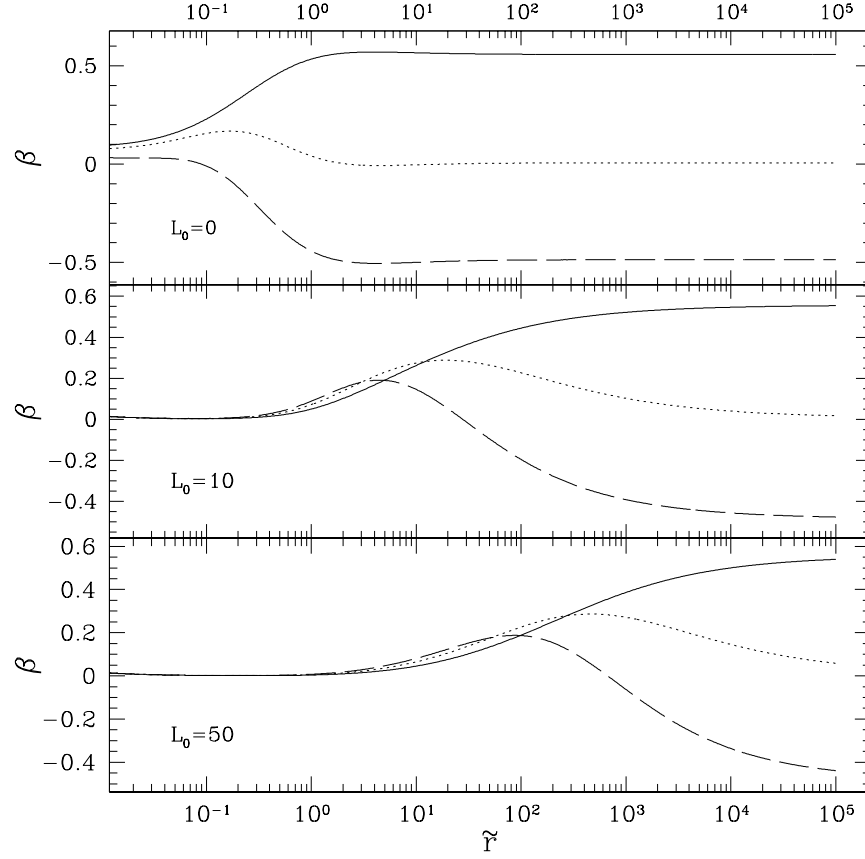


Figure 4.33: Anisotropy profiles of GCSs directly tracing a $\gamma = 1$ Dehnen model host galaxy, with circularity function $j(h) = 0.5[1 + h \sin(ah)]$. In descending order each panel has $L_0 = 0, 10, 50$, and in every panel are displayed the anisotropy profiles generated with $a = -2, -3, -4$ (solid, dotted, curved respectively).

properties, except that the values of a have been made negative in the latter case. Comparison of these panels reveals that the order of the anisotropies resulting from the different values of a have been reversed, i.e. $a = -2$ (solid) gives radial anisotropy, $a = -3$ (dotted) gives very slight radial anisotropy, and $a = -4$ (dashed) gives tangential anisotropy. This is unsurprising given that the average gradient of the circularity function is given by $\frac{\Delta j}{\Delta h} = 0.5 \sin(a)$, and thus changing the sign on a will just change the sign of $\frac{\Delta j}{\Delta h}$. The reason for the resulting asymptotic anisotropies being about twice

as strong for $a = -2, -3, -4$ is much less obvious, though it is likely to do with the fact that $j(h)$ is typically smaller when a is negative. This is because for positive a the circularity function initially increases from 0.5 towards a peak, and provided $a > 2.02876$ thereafter begins descending, whereas with a negative the circularity function initially decreases from 0.5 towards a trough, and provided $a < -2.02876$ thereafter begins ascending. Consequently the integration over $j(h)$ in equations (3.54) and (3.55) will weight the velocity distributions as functions of energy differently, resulting in a different anisotropy profile. Since the integration over $\tilde{g}(\tilde{E})j(h)$ must equate to the tracer number density, the normalisation of $j(h)$ is arbitrary since $\tilde{g}(\tilde{E})$ must always compensate. Therefore, what is important here is the relative change in $j(h)$. The non-zero stationary points in $j(h)$ occur at $h = -2.02876/a$ and $2.02876/a$, which both correspond to $j(h_{stat}) = 0.5[1 + 1.81971/a]$. Therefore, the relative change in the circularity function at this point is $j(h_{stat})/j(0) = 1 + 1.81971/a$, which for example with $a = 3$ gives $j(h_{stat})/j(0) = 1.60657$, whereas $a = -3$ gives $j(h_{stat})/j(0) = 0.39343$ (note that since we are assuming that $h = h_{stat}$, we must have $a > 2.02876$ or $a < -2.02876$ for the stationary point to lie in the interval $0 \leq h \leq 1$, and therefore $j(h_{stat})/j(0) < 0$ is not possible). Hence, when $a = -3$ is used instead of $a = 3$, the relative change in $j(h)$ goes from a factor of $\simeq 1.6$ to a factor of $1/0.39343 \simeq 2.5$. The only way for these relative changes with positive and negative a to converge is for $|a| \rightarrow \infty$, and therefore we would expect that for very large $|a|$, only the sign of the asymptotic value of the anisotropy should change when $-|a|$ is used instead. On the other hand, $|a| \rightarrow \infty$ will result in a tiny relative change, and therefore will correspond to very weak anisotropy. This was confirmed by producing anisotropy profiles with $a = -1000, 1000$ in the circularity function, resulting in asymptotic anisotropy values of $\beta = -0.002389, 0.002387$.

The middle and bottom panels of Figure 4.33 have identical properties to the top panel (directly tracing a $\gamma = 1$ Dehnen model host galaxy with $j(h) = 0.5[1 + h \sin(ah)]$, $a = -2, -3, -4$), except that the middle panel has $\tilde{L}_0 = 10$, and the bottom panel has $\tilde{L}_0 = 50$. The most obvious feature of the anisotropy profiles produced with non-zero \tilde{L}_0 is that more of the inner region becomes isotropic with bigger values

of \tilde{L}_0 . The reason for this is that in the central regions, \tilde{E} is relatively small, and therefore $\tilde{L} \leq \tilde{L}_c(\tilde{E}) \ll \tilde{L}_0$, and therefore $h \simeq 0$. Consequently, the distribution function becomes $\tilde{f}(\tilde{E}, \tilde{L}) = \tilde{g}(\tilde{E})j(0)$, i.e. the distribution function only depends on energy, and is therefore isotropic. Furthermore, the larger the value of \tilde{L}_0 , the more horizontally stretched the resulting anisotropy profile (the same can be seen by comparing Figures 4.27, 4.28, 4.29, 4.30 to 4.23, 4.24, 4.25 and 4.26). The reason for this is due to the fact that \tilde{L}_0 always appears near to $\tilde{L}_c(\tilde{E})$. In order for the behaviour of the anisotropy profile to be similar to that in the $\tilde{L}_0 = 0$ case, we must have that $\tilde{L}_c(\tilde{E}) \gg \tilde{L}_0$. For larger \tilde{L}_0 , clearly this can only happen at large \tilde{E} . In general we have that $\tilde{L}_c^2 = \tilde{r}^3 \frac{d\tilde{\Phi}}{d\tilde{r}}$, and so in this particular case $\tilde{L}_c^2 = 2^{4-\gamma} P_0 \tilde{r}^{4-\gamma} / (3-\gamma)(1+\tilde{r})^{3-\gamma}$, asymptotically this becomes $\tilde{L}_c^2 = 2^{4-\gamma} P_0 \tilde{r} / (3-\gamma)$, and so $\tilde{L}_c \propto \sqrt{\tilde{r}}$. Consequently a large factor in \tilde{r} is required before $\tilde{L}_c(\tilde{E}) \gg \tilde{L}_0$ (e.g. Gerhard 1991). Therefore, asymptotically the behaviour of anisotropy profiles generated with the same a are very similar if not identical, regardless of what value \tilde{L}_0 takes.

In conclusion, the behaviour of anisotropy profiles resulting from different circularity functions is basically well understood in terms of $\frac{dj}{dh}$ and the relative change in $j(h)$, and the dependence on L_0 in terms of the behaviour of $\tilde{L}_c(\tilde{E})$. The procedure for taking simulated initial GCSs such as those presented in this Chapter and evolving them through time is described next in Chapter 5. Following this, demonstrations are given of the application of the time evolution procedure in Chapter 6.

5 Dynamical Evolution of Simulated GCSs

Once the initial tracer system has been set up according to the procedure in Chapter 3, it remains to evolve the system through time with dynamical effects such as evaporation, shocking, stellar evolution, and dynamical friction acting on the simulated GCs. The equations of motion for a particle orbiting in a general host gravitational potential are derived, including the effects of orbital decay due to dynamical friction in section 5.1. Following this, a thorough literature review is conducted, drawing together prescriptions for evaporation mass-loss rate, tidal shocking mass-loss rate, mass-loss due to stellar evolution, and the magnitude of dynamical friction in section 5.2, to be applied during the integration of the equations of motion. This Chapter is then concluded with a description of the sequential application of the physics of sections 5.1 and 5.2 when applying this method to evolve a tracer system and find its final mass function, and spatial and kinematic distributions.

5.1 Equations of Motion

As the system is spherically symmetric, the inclinations of tracer orbits are unimportant and the equations of motion do not have any dependence on γ_{inc} . Instead, the equations of motion describe the motion of each tracer on its orbital plane, depending only on galactocentric distance and radial and tangential speeds.

Defining $\tilde{\mathbf{r}}_M$ to be the position vector, and $\tilde{\mathbf{v}}_M$ to be the velocity of a tracer, then \tilde{r}_M is the magnitude of $\tilde{\mathbf{r}}_M$, and $\tilde{v}_M = \sqrt{\tilde{v}_t^2 + \tilde{v}_r^2}$ is the magnitude of the velocity. Using the position and velocity vectors to define a plane, $\tilde{\mathbf{L}} = \tilde{\mathbf{r}}_M \times \tilde{\mathbf{v}}_M$ is the orbital angular momentum vector perpendicular to this plane (the $\hat{\mathbf{z}}$ direction), and \tilde{L} is its magnitude. The tangential and radial speeds are given by equations (3.44) and (3.48),

repeated here for ease of reference:

$$\tilde{v}_r^2 = 2 \left[\tilde{E} - \tilde{\Phi}(\tilde{r}_M) \right] - \frac{\tilde{L}^2}{\tilde{r}_M^2} \quad (5.1)$$

$$\tilde{v}_t^2 = \frac{\tilde{L}^2}{\tilde{r}_M^2} \quad (5.2)$$

The equations of motion are obtained by taking the time derivatives of these equations, leading to:

$$\dot{\tilde{v}}_r = \frac{1}{\tilde{v}_r} \left[\frac{d\tilde{E}}{d\tilde{t}} - \frac{\tilde{v}_t}{\tilde{r}_M} \frac{d\tilde{L}}{d\tilde{t}} \right] + \frac{\tilde{v}_t^2}{\tilde{r}_M} - \frac{d\tilde{\Phi}}{d\tilde{r}} \bigg|_{\tilde{r}_M} \quad (5.3)$$

$$\dot{\tilde{v}}_t = \frac{1}{\tilde{r}_M} \frac{d\tilde{L}}{d\tilde{t}} - \frac{\tilde{v}_t \tilde{v}_r}{\tilde{r}_M} \quad (5.4)$$

For an object undergoing dynamical friction, $d\tilde{E}/d\tilde{t} \leq 0$ and $d\tilde{L}/d\tilde{t} \leq 0$. Otherwise, $d\tilde{E}/d\tilde{t} = d\tilde{L}/d\tilde{t} = 0$, and equations (5.3) and (5.4) reduce to the better known standard equations of motion (e.g. Binney & Tremaine 2008). Defining \hat{z} to be the unit vector normal to the orbital plane, the effects of dynamical friction may be quantified by considering the power and torque acting on an orbiting body (e.g. van den Bosch et al. 1999):

$$\frac{d\tilde{E}}{d\tilde{t}} = \tilde{\mathbf{a}}_{df} \cdot \tilde{\mathbf{v}}_M = \tilde{a}_{df} \tilde{v}_M \cos \kappa \quad (5.5)$$

$$\frac{d\tilde{L}}{d\tilde{t}} = (\tilde{\mathbf{r}}_M \times \tilde{\mathbf{a}}_{df}) \cdot \hat{\mathbf{z}} = -\tilde{r}_M \tilde{a}_{df} \sin \alpha \quad (5.6)$$

where $\tilde{\mathbf{a}}_{df}$ is the dynamical friction vector, and \tilde{a}_{df} is its magnitude, the deceleration due to dynamical friction. κ is the angle between the velocity and dynamical friction vectors, and α is the angle between the position and dynamical friction vectors. As dynamical friction is always antiparallel to velocity, $\kappa = \pi$ radians, and $\sin \alpha = \tilde{v}_t/\tilde{v}_M$. Thus:

$$\frac{d\tilde{E}}{d\tilde{t}} = -\tilde{a}_{df}\tilde{v}_M \quad (5.7)$$

$$\frac{d\tilde{L}}{d\tilde{t}} = -\tilde{a}_{df}\frac{\tilde{r}_M\tilde{v}_t}{\tilde{v}_M} \quad (5.8)$$

$$\ddot{\tilde{r}}_M = \frac{\tilde{v}_t^2}{\tilde{r}_M} - \frac{d\tilde{\Phi}}{d\tilde{r}} - \tilde{a}_{df}\frac{\dot{\tilde{r}}_M}{\tilde{v}_M} \quad (5.9)$$

$$\dot{\tilde{v}}_t = -\frac{\tilde{v}_t\dot{\tilde{r}}_M}{\tilde{r}_M} - \tilde{a}_{df}\frac{\tilde{v}_t}{\tilde{v}_M} \quad (5.10)$$

A choice of \tilde{a}_{df} is then selected from the prescriptions (see section 5.2.4), and the equations of motion are integrated to give $\tilde{v}_r(\tilde{t})$, $\tilde{v}_t(\tilde{t})$ and $\tilde{r}_M(\tilde{t})$. These time-dependent quantities can then be used when calculating the rate of mass-loss due to the selected prescriptions (see section 5.2), allowing simultaneously solving for $\widetilde{M}_c(\tilde{t})$ also. Thus the orbit of every object in the system is integrated to the desired age, and the evolved GCMF, evolved tracer spatial density, and evolved kinematic distribution can be built from the surviving tracer GCs.

5.2 Prescriptions

5.2.1 Evaporation

Evaporation is broken down into two regimes, the energy criterion and the apocentre criterion. The difference between these regimes is the criterion a star must meet before being considered to have escaped. The energy criterion only requires that a star has sufficient energy to escape before it is assumed to leave its host cluster, whereas the apocentre criterion requires a star to have both sufficient energy to escape, and reach the tidal boundary before it is assumed to leave its host cluster. During transit to the tidal boundary, a star with sufficient energy to escape may interact with another star, and have its energy reduced such that escape is no longer possible. This effect manifests in the mass-loss rate through an additional dependence on mass. Whichever of these categories a mass-loss rate may belong to, it may always be written with the

same basic scalings (equation 1.25), repeated here:

$$\frac{d\widetilde{M}_c}{d\tilde{t}} = \mu_{ev} \propto -M_c^{1-x} \rho_h^{1/2} f\left(\frac{\rho_t(r_{ev})}{\rho_h}\right) \quad (5.11)$$

where $x = 1$ for the energy criterion, and $0 \leq x < 1$ for the apocentre criterion, r_{ev} is the galactocentric distance at which the rate of mass-loss matches the time averaged mass-loss rate, incorporating mass-loss rate dependence on orbital shape, and $f(\rho_t/\rho_h)$ incorporates mass-loss rate dependence on cluster internal structure/degree of Roche lobe filling (see Chapter 1 for a more detailed discussion). Each evaporation prescription is given in Table 5.3 in terms of the uncertainties represented in equation (1.25) for easy comparison.

5.2.1.1 Energy Criterion

King (1966) investigated the escape rate of the single mass family of cluster models derived in the study in a point mass galaxy, taking the effective tides determining the rate of mass-loss to be set at pericentre. King (1966) found that to $\sim 20\%$, every part of a King model cluster loses the same fraction of stars per unit time (this becomes more precise for higher W_0). Additionally, King (1966) found that when omitting mass-loss due to stellar evolution and tidal shocks, as a King model cluster loses mass due to evaporation, it becomes more concentrated and evolves along the sequence of King models. In the dimensionless notation introduced in Chapter 3, the King (1966) mass-loss rate can be expressed as:

$$\frac{d\widetilde{M}_c}{d\tilde{t}} = -\frac{27}{8} \left(\frac{P_0 \tilde{\rho}_t}{6\pi}\right)^{1/2} \ln(0.5\widetilde{M}_c) F(W_0) \quad (5.12)$$

where $F(W_0)$ is a weak function of W_0 , with a maximum of $F(W_0) \simeq 8$ and a minimum of $F(W_0) \simeq 3$ over the range $W_0 \in [2.5, 10]$. This function $F(W_0)$ is essentially f , describing mass-loss rate dependence on cluster structure. Note also that the expression derived by King (1966) had a Coulomb logarithm depending on the number of stars in the core, but that this has been approximated to \widetilde{M}_c in equation (5.12). As this is only

a logarithmic dependence, this approximation is not likely to amount to a difference larger than a factor of a few.

Vesperini & Heggie (1997) performed a series of N-body simulations of clusters to estimate mass-loss due to stellar evolution, evaporation (both based on Chernoff & Weinberg 1990), and disk shocking (based on Chernoff, Kochanek & Shapiro 1986). They assumed a point mass host galaxy with a disk, and modelled all clusters to be on circular orbits with rotation speeds of 220 km s^{-1} that always crossed the disk perpendicularly. As they assumed that all cluster orbits were circular, they avoided having to address any mass-loss rate dependence on orbital shape (i.e. they did not consider any dependence on \tilde{r}_{ev}). Consequently, their only time dependent variable was cluster mass, expressed as:

$$\frac{M_c(t)}{M_{c,i}} = 1 - \frac{0.828}{F_{CW}} \frac{t}{\text{Gyr}} \quad (5.13)$$

where M_i is initial cluster mass, and F_{CW} is a parameter adopted from Chernoff & Weinberg (1990). Thus, by substituting in F_{CW} and differentiating, their mass-loss rate can be expressed in dimensionless notation as:

$$\frac{d\widetilde{M}_c}{d\tilde{t}} = -\frac{92}{75} \left(\frac{M_\odot}{\bar{m}_0} \right) \ln(\widetilde{M}_{c,i}) (P_0 \tilde{\rho}_t)^{1/2} \quad (5.14)$$

where $M_\odot/\bar{m}_0 \simeq 1$ and hence is ignored. Furthermore, since they assumed that all clusters were tidally limited and did not consider any mass-loss rate dependence on cluster structure, they implicitly set $f = \text{const}$.

Fall & Zhang (2001) modelled the evolution of a GCMF including evaporation, tidal shocking, and stellar evolution in an isothermal disk galaxy. For clusters on elliptical orbits, they assumed that the tidal radii of their clusters were imposed at pericentre (i.e. $\tilde{r}_{ev} = \tilde{r}_p$), following Innanen, Harris & Webbink (1983). Furthermore, they assumed Hénon (1961) self-similar evolution for all of their clusters, such that the fractional mass-loss per relaxation time is given by $\xi = 0.045$, and $\tilde{r}_h = 0.145\tilde{r}_t$. Furthermore they ignored any variation in the mean stellar mass or in the Coulomb logarithm by setting $\Lambda = 12$. In dimensionless notation, their mass-loss rate due to

evaporation is:

$$\frac{d\widetilde{M}_c}{d\tilde{t}} = -269\xi \left(\frac{P_0\tilde{\rho}_t}{4\pi} \right)^{1/2} \bar{\tilde{m}} \ln(\Lambda) \quad (5.15)$$

since they did not model evolving average stellar mass, $\bar{\tilde{m}} = 1$. Furthermore, by assuming self-similar models with $\tilde{r}_h = 0.145\tilde{r}_t$, cluster structure is always just a rescaled version of itself. Consequently, as the cluster evolves it will always occupy the same fraction of its Roche lobe, and therefore this treatment does not explicitly account for any dependence on cluster structure.

Jordán et al. (2007) developed simple models describing GCMF evolution assuming evaporation dominated mass-loss from a Schechter (1976) CIMF. By fitting these models to the observed GCMFs of Virgo galaxies, they estimated a fitting parameter $C_J \simeq 840 \pm 560$ directly related to the average rate of mass-loss in GCs (where $dM_c/dt = -C\rho_h^{1/2}$ defines C , see Chapter 2). In dimensionless notation the mass-loss rate due to evaporation for each cluster is given by:

$$\begin{aligned} \frac{d\widetilde{M}_c}{d\tilde{t}} &= -\tilde{C} \left(\frac{P_0\tilde{\rho}_h}{3} \right)^{1/2} \\ \tilde{C} &= \frac{C}{\bar{\tilde{m}}} \left(\frac{3}{8\pi G} \right)^{1/2} \end{aligned} \quad (5.16)$$

such that for each study done to obtain C , an accompanying estimate of $\bar{\tilde{m}}$ is required to define \tilde{C} . By averaging over entire GCSs in this way, \tilde{C} absorbs the uncertainties in evaporation mass-loss rate in equation (1.25), such as dependence on cluster structure, orbital shape, and apocentre/energy escape criterion, such that accounting for these uncertainties explicitly is not required. However, due to a lack of explicit dependence on these additional considerations, this formula would predict the same rate of mass-loss for any clusters with the same $\tilde{\rho}_h$, regardless of orbit/structure, etc. Thus, attempting to predict the evaporation mass-loss rate for a single cluster using C will probably not be closer than a factor of a few to the true value. However, when applied instead to a large sample of clusters with a range of structures and orbits, mass-loss rates will be underestimated for some clusters and overestimated for others, with the overall

average mass-loss rate relatively well reproduced (if Virgo GCSs are representative, which seems likely).

McLaughlin & Fall (2008) refined the approach of Jordán et al. (2007) in the Milky Way by treating the mass of each cluster as a time- and density-dependent probability distribution, and the GCMF as the sum of each of these mass probability distributions. Then, the half-mass density of each individual cluster is used in estimating C directly, rather than fitting an average mass lost per cluster to all clusters in the sample simultaneously and inferring C using the average cluster ρ_h . With this method, McLaughlin & Fall (2008) obtained $C_{MF} = 1100$. Similarly, Chandar, Fall & McLaughlin (2007) used individual cluster densities in M104 to estimate C directly, obtaining a value of $C_{CFM} = 560$. The same method was used in Chapter 2 in 22 different Virgo galaxies to obtain an average $C_{virgo} = 810 \pm 170$. Furthermore, Goudfrooij (2012) used the same technique applied to intermediate age clusters in NGC1316 to estimate $C_G = 875$.

By adapting the mass-loss formula of Wielen (1988) and utilising the results of other authors, Baumgardt (1998) derived a mass-loss rate for clusters in a point mass gravitational potential incorporating dependence on the degree of Roche lobe under-filling/cluster structure, obtaining (in dimensionless notation):

$$\frac{d\widetilde{M}_c}{d\widetilde{t}} = -\xi_0 \sqrt{1 + \left(\alpha \frac{\widetilde{r}_h}{\widetilde{r}_t} \right)^3} \left(\frac{2P_0 \widetilde{\rho}_h}{3} \right)^{1/2} \frac{\ln(0.4\widetilde{M}_c)}{0.138} \quad (5.17)$$

where ξ_0 is the fraction of mass lost per relaxation time for a cluster in isolation estimated by Aarseth & Heggie (1993) to be $\xi_0 \simeq 0.016$. For a cluster in isolation, $\widetilde{r}_h/\widetilde{r}_t \ll 1$, and $d\widetilde{M}_c/d\widetilde{t} \propto -\xi_0 \widetilde{\rho}_h^{1/2}$, as expected (e.g. Spitzer 1987). On the other hand, for a tidally limited cluster, $\alpha(\widetilde{r}_h/\widetilde{r}_t)^3 \gg 1$ (assuming $\alpha > 1$), making $d\widetilde{M}_c/d\widetilde{t} \propto -0.5\xi_0 \alpha^{1/2} \widetilde{\rho}_t^{1/2}$. Thus, α is a fitting parameter, and was estimated from the results of Lee, Fahlman & Richer (1991), Lee & Goodman (1995) and de La Fuente Marcos (1995) to be $\alpha \simeq 14.9$. Therefore, in this treatment $f = \sqrt{1 + \alpha(\widetilde{r}_h/\widetilde{r}_t)^3} = \sqrt{1 + \alpha \widetilde{\rho}_t/2\widetilde{\rho}_h}$.

Gieles & Baumgardt (2008) also derived mass-loss rates in a point mass gravitational potential depending on the degree of Roche lobe under-filling and cluster

structure, denoted $\mathcal{R} = \tilde{r}_h/\tilde{r}_t$ in a similar way to Baumgardt (1998). They accomplished this by assuming a Gaussian velocity distribution for a range of clusters with different \mathcal{R} , and calculated the fraction of stars with velocities greater than the escape velocity of an isolated cluster, finding that $\xi = \xi_0 \exp(10\mathcal{R})$ is a good representation. Severely under-filling (i.e. isolated) clusters are then represented by $\mathcal{R} \simeq 0$, and thus ξ_0 is the escape fraction per relaxation time for isolated clusters, with an adopted value of $\xi_0 = 0.0074$ (Spitzer 1987). Furthermore, they set $\lambda = 0.11$. Consequently, in this treatment $f = \exp(10\tilde{r}_h/\tilde{r}_t) = \exp(7.94\sqrt[3]{\tilde{\rho}_t/\tilde{\rho}_h})$. In dimensionless notation, Gieles & Baumgardt (2008) found the mass-loss rate to be:

$$\frac{d\tilde{M}_c}{d\tilde{t}} = -\frac{\xi_0 \exp(10\tilde{r}_h/\tilde{r}_t) \ln(\lambda\tilde{M}_c)}{0.138} \left(\frac{2P_0\tilde{\rho}_h}{3}\right)^{1/2} \quad (5.18)$$

Gieles & Baumgardt (2008) claimed that for $r_h/r_t > 0.05$, which they called the ‘tidal regime’, $\exp(10r_h/r_t) \propto (r_h/0.05r_t)^{3/2}$, and that therefore their mass-loss rate will scale with $\tilde{\rho}_h^{1/2}$ in the isolated regime ($r_h/r_t < 0.05$), after which the $\tilde{\rho}_h$ dependence will cancel and the mass-loss rate will instead scale as $\tilde{\rho}_t^{1/2}$.

Gieles et al. (2011) analytically derived a model for evolving clusters from steadily expanding clusters in isolation (based on Hénon 1965) to homologously contracting under evaporation (based on Hénon 1961) in the tidal field of an isothermal sphere. Additionally, they set $\ln(0.02N)$ with $N = 10^5$ in place of a varying Coulomb logarithm. In dimensionless notation, their energy criterion mass-loss rate is:

$$\frac{d\tilde{M}_c}{d\tilde{t}} = -100\zeta\bar{\bar{m}} \left(\frac{P_0}{8\pi\alpha^3}\right)^{1/2} \tilde{\rho}_t^{1/2} \quad (5.19)$$

where ζ is the constant fraction of energy expended by a cluster on its evolution per relaxation time (estimated to be $\zeta \simeq 0.2$; Gieles, Baumgardt & Heggie 2010, see also section 1.4.4), $\bar{\bar{m}}$ is the dimensionless average stellar mass, and α is the final ratio of \tilde{r}_h/\tilde{r}_t . Since these models assumed homologous evolution in later stages of cluster evolution, $\tilde{r}_h/\tilde{r}_t = \alpha = 0.145$ (Hénon 1961), this is again equivalent to setting $f = \text{const}$, as homologous clusters will always occupy the same fraction of their Roche lobe. As Gieles et al. (2011) treat $\bar{\bar{m}}$ as a constant in their models, the value of $\bar{\bar{m}}$ is just unity.

5.2.1.2 Apocentre Criterion

Heggie et al. (1998) compared results of star cluster evolution simulations obtained using different methods but with the same initial conditions to check for consistency. Once the experiment had been run and the results analysed, a number of problems became apparent (Fukushige & Heggie 2000). As running N-body simulations with a realistic number of stars ($N \sim 10^5$) take a considerable amount of time, simulations were run using a smaller number and the results scaled appropriately. However, when this was done, it was found that the cluster lifetimes displayed a significant dependence on N , rather than being independent of N as had been assumed would be the case. The cause of this was attributed to the way that escaping stars had been dealt with in the simulations, namely the energy criterion versus the apocentre criterion. Under the energy criterion, as soon as a star achieves escape velocity it is removed from the simulation. However, realistically once a star has achieved escape velocity, it must then reach the edge of the cluster before actually escaping, known as the apocentre criterion. This difference had been pointed out previously by Chandrasekhar (1942) and King (1959). This more strict condition makes a difference because while travelling to the cluster edge, the potential escaper may encounter another star and have its velocity down-scattered below escape velocity, such that it will remain bound to the cluster (see Chapter 1). Fukushige & Heggie (2000) predicted that the time-scale for escape of a star with sufficient energy to escape scales as $t_{esc} \propto \hat{E}^{-2}$, where \hat{E} is the excess energy beyond the minimum required for escape. Using this result, Baumgardt (2001) derived models of potential escaper populations in equilibrium (i.e. as many new stars obtaining sufficient energy to escape as are actually leaving the cluster per unit time), and predicted that rather than the mass-loss being a fraction ξ of the mass per relaxation time, it is a constant fraction per $t_{rh}^x t_{cr}^{1-x}$. Ignoring the weakly varying Coulomb logarithm, the scalings of relaxation time and crossing time ($t_{rh} \propto M_c \rho_h^{-1/2}$, $t_{cr} \propto \rho_h^{-1/2}$) can be used to show that:

$$\frac{dM_c}{dt} \propto -M_c^{1-x} \rho_h^{1/2}$$

which for $x \neq 1$ introduces dependence on cluster mass.

Gieles & Baumgardt (2008) derived models including this effect in the same way as their energy criterion models, obtaining:

$$\frac{d\widetilde{M}_c}{d\widetilde{t}} = -\xi_0 \exp(10\widetilde{r}_h/\widetilde{r}_t) \left[\frac{\ln(\lambda\widetilde{M}_c)}{0.138} \right]^{3/4} \left(\frac{\widetilde{M}_c}{k} \right)^{1/4} \left(\frac{2P_0\widetilde{\rho}_h}{3} \right)^{1/2} \quad (5.20)$$

where k is a constant depending on cluster structure, with $k = 3.85$ for a $W_0 = 5$ King (1966) cluster. In terms of the scalings in equation (1.25), they have $x = 0.75$ and $f = \exp(10\widetilde{r}_h/\widetilde{r}_t) = \exp(7.94\sqrt[3]{\widetilde{\rho}_t/\widetilde{\rho}_h})$.

Gieles et al. (2011) derive models in an appendix accounting for mass-loss under the apocentre criterion. Their other assumptions about cluster evolution remained the same as in their derivation of mass-loss rates under the energy criterion. These were that clusters steadily expand from isolation (based on Hénon 1965) to homologously contracting clusters undergoing evaporation in a tidal field (based on Hénon 1961). Additionally, they set $\ln(\Lambda) = \ln(0.02N)$ with $N = 10^5$ in place of a varying Coulomb logarithm. In dimensionless notation, their apocentre criterion mass-loss rate is:

$$\frac{d\widetilde{M}_c}{d\widetilde{t}} = -100\zeta \left(\frac{P_0}{8\pi\alpha^3} \right)^{1/2} \widetilde{\rho}_t^{1/2} \left(\frac{\widetilde{M}_c}{\widetilde{M}_{c,1}} \right)^{1/4} \quad (5.21)$$

where $\widetilde{M}_{c,1} \simeq 10^5$ is a fitting parameter. Again since late-stage cluster evolution is assumed to be homologous, cluster structure is always just a rescaled version of itself, and will always occupy the same fraction of its Roche lobe. Consequently, this treatment does not explicitly account for any dependence on cluster structure. Furthermore, they found that $x = 0.75$ gives the best fits to their N-body data.

Baumgardt & Makino (2003) analysed mass-loss due to stars escaping under the apocentre criterion for a range of King models using N-body simulations. They found that $x \simeq 0.75$ for $W_0 = 5$, and that $x \simeq 0.82$ for $W_0 = 7$, and concluded that x may vary

weakly with cluster structure. However, this study focused on mass-loss depending on cluster mass, without any explicit dependence on cluster structure/degree of Roche lobe filling. Therefore, these different values of x for different W_0 may actually be reflecting degeneracy between x and f .

Further studies have claimed that in fact x depends on the strength of the tidal field (Tanikawa & Fukushige 2005), and on the mass profile of the host galaxy more than on cluster structure (Tanikawa & Fukushige 2010). Combining results from both studies, Tanikawa & Fukushige (2010) derive for the mass-loss time-scale:

$$\tilde{t}_{mloss}(\hat{r}_t, \alpha) \simeq 254 t_{1000} [1 + f(\alpha) g(\hat{r}_t)] (P_0 \tilde{\rho}_{h,i})^{-1/2} \left[1.5 \times 10^{-5} \widetilde{M}_{c,i} \right]^{x(\hat{r}_t)} \quad (5.22)$$

$$\hat{r}_t = \frac{\tilde{r}_{t,i}}{\tilde{r}_{h,i}} \quad (5.23)$$

$$\alpha = 1 + P_0 \frac{\tilde{\rho}}{\tilde{\Omega}^2} \quad (5.24)$$

where α is a parameter encapsulating the mass profile of the host galaxy, and \hat{r}_t depends on both the cluster structure and the strength of the tidal field, with a subscript i indicating the initial value. t_{1000} is a normalisation parameter based on clusters with initial relaxation times equal to 1000 N-body time units in their simulations. In their N-body results, Tanikawa & Fukushige (2010) also fit for the functions $t_{1000}(\hat{r}_t)$, $x(\hat{r}_t)$, $f(\alpha)$, and $g(\hat{r}_t)$ to obtain:

$$t_{1000}(\hat{r}_t) \simeq 2 \times 10^4 \frac{\hat{r}_t^4}{\hat{r}_t^4 + 2 \times 10^3} \quad (5.25)$$

$$x(\hat{r}_t) \simeq \frac{\hat{r}_t^4}{\hat{r}_t^4 + 1 \times 10^2} \quad (5.26)$$

$$f(\alpha) \simeq \frac{15}{16 - \alpha^2} \quad (5.27)$$

$$g(\hat{r}_t) = \begin{cases} 1 & \text{for } \hat{r}_t \lesssim 8 \\ 0 & \text{for } \hat{r}_t \gtrsim 8 \end{cases} \quad (5.28)$$

where t_{1000} behaves such that in the limit of very strong tides, cluster mass-loss is virtually instantaneous, i.e. $t_{1000} \rightarrow 0$ as $\hat{r} \rightarrow 0$. On the other hand, in the absence of a tidal field, mass-loss occurs on a time-scale expected for an isolated cluster, i.e. $t_{1000} \rightarrow 2 \times 10^4$ as $\hat{r} \rightarrow \infty$. Their function x is designed such that there is no mass

dependence in the mass-loss time-scale in the limit of strong tides, i.e. $x \rightarrow 0$ as $\hat{r} \rightarrow 0$, but approaches unity for a cluster in isolation, $x \rightarrow 1$ as $\hat{r} \rightarrow \infty$. If the local density is written as $\tilde{\rho} = \tilde{r}^{-k}$, then $\alpha = 4 - k$. Then given that all systems must have density gradients bounded between that of a point mass ($k = 3$), and a homogeneous sphere ($k = 0$), this means that $\alpha \in [1, 4]$. Then as is expected for a homogeneous sphere, $f(\alpha \rightarrow 4) \rightarrow \infty$, i.e. tides are ineffective in a homogeneous sphere. Bringing all of these equations together, in dimensionless notation their mass-loss rate is given by:

$$\frac{d\widetilde{M}_c}{d\tilde{t}} = -\frac{\widetilde{M}_{c,i}}{\tilde{t}_{mloss}} \quad (5.29)$$

Although this appears rather complicated, in terms of equation (1.25) this treatment of mass-loss rate depends on (initial) cluster structure through $\hat{r} = \tilde{r}_{t,i}/\tilde{r}_{h,i}$, and on (initial) cluster mass through \widetilde{M}_c^{1-x} with $x = \hat{r}_t^4/(\hat{r}_t^4 + 100)$. The remaining terms depend on the host galaxy, and are therefore constant during cluster dynamical evolution. Consequently this mass-loss rate depends only on initial cluster parameters, unlike the mass-loss rates of other authors, and is therefore constant during cluster evolution. Although this will be less realistic, it does have the advantage of making the integrations much simpler.

Lamers, Baumgardt & Gieles (2010) attempted to go further still, investigating how mass-loss rate depends on the degree of Roche lobe filling, on the orbital ellipticity, on a varying stellar mass spectrum (including different metallicities), in addition to internal cluster structure. Additionally, rather than ignore any weak dependence on the Coulomb logarithm, they approximate $N/\ln(0.02N) \simeq N^{0.8}$ for $10^4 < N < 10^6$ in their derivations. Their investigation used the N-body results of Baumgardt & Makino (2003) for Roche lobe filling simulations (note that the host galaxy potential in these simulations was only modelled as a singular isothermal sphere), supplemented by their own simulations, which were presumably also for a singular isothermal sphere host galaxy. They broke the mass-loss up into four sections: direct stellar evolution mass-loss, stellar evolution induced mass-loss, and pre- and post-core-collapse evaporation (pre-cc and post-cc). Only the pre-cc and post-cc evaporation mass-loss is discussed here, as mass-loss related to stellar evolution is discussed in section 5.2.3. Their equa-

tion for mass-loss is:

$$\frac{dM_c}{dt} = -\frac{M_c^{1-\delta}}{At_L} \quad (5.30)$$

where $\delta \simeq [0.65, 0.85]$, depending on the degree of Roche lobe filling, internal cluster structure, and whether the mass-loss is pre-cc or post-cc, and A is a constant depending on whether mass-loss is pre-cc or post-cc. Their t_L is given by:

$$\frac{t_L}{\text{Myr}} = \frac{t_{ref}^N}{\text{Myr}} \left(\frac{\bar{m}}{M_\odot} \right)^{-\delta} \left((1-e) \frac{R_{gal}}{8.5 \text{ kpc}} \frac{220 \text{ km s}^{-1}}{v_{gal}} \right) \quad (5.31)$$

$$\bar{m} = a (t_0 M_{c,i}^\delta)^b \mathcal{F}_7^c \quad (5.32)$$

where t_{ref}^N , a , b and c are fitting parameters deduced through N-body results, \bar{m} is the average stellar mass, e is the orbital ellipticity, R_{gal} and v_{gal} are the galactocentric distance and galaxy circular speed, and $M_{c,i}$ is the cluster initial mass. The degree of Roche lobe under-filling is specified by $\mathcal{F}_{W_0} = \tilde{r}_{t,W_0}/\tilde{r}_J$, the ratio of zero density radius to Jacobi radius (i.e. $\mathcal{F}_{W_0} \leq 1$), where the subscript W_0 denotes the value of the King model central potential, and thus \mathcal{F}_{W_0} is a function of \tilde{r}_J only. For example $W_0 = 5, 7$ King models have $\tilde{r}_{t,5} = 10.7$, $\tilde{r}_{t,7} = 33.71$, and it is therefore possible to write $\mathcal{F}_5 = \tilde{r}_{t,5}/\tilde{r}_J = \mathcal{F}_7 \tilde{r}_{t,5}/\tilde{r}_{t,7} = \mathcal{F}_7/3.15$, which can be used to simplify some of the Lamers, Baumgardt & Gieles (2010) equations. Combining all of this, the Lamers, Baumgardt & Gieles (2010) mass-loss rate for all but post-cc Roche lobe under-filling clusters can be expressed in dimensionless notation as:

$$\frac{d\widetilde{M}_c}{d\tilde{t}} = - \left(\frac{\bar{m}_0}{M_\odot} \right)^{\eta(bq\delta-1)} \frac{a^{q\eta} \widetilde{M}_{c,i}^{bq\eta\delta}}{\left[\frac{9}{340} \tilde{t}_{ref}^N (1-e) \right]^q} \widetilde{M}_c^{1-\eta} \left(\frac{P_0 \tilde{\rho}_t}{6} \right)^{q/2} \quad (5.33)$$

$$q = \frac{1}{1+b\eta} \quad (5.34)$$

mass-loss for post-cc Roche lobe under-filling clusters is instead given by:

$$\frac{d\widetilde{M}_c}{d\tilde{t}} = - \left(\frac{\bar{m}_0}{M_\odot} \right)^{\gamma-\eta} \frac{a_{cc}^\eta a^{q\delta} \widetilde{M}_{c,i}^\gamma}{6.48 A \mathcal{F}_7^{0.127+c_{cc}+qc\delta} \left[\frac{9}{340} \tilde{t}_{ref}^N (1-e) \right]^q} \widetilde{M}_c^{1-\eta} \left(\frac{P_0 \tilde{\rho}_t}{6} \right)^{q/2} \quad (5.35)$$

$$q = \frac{1}{1+b\delta} \quad (5.36)$$

$$\gamma = b_{cc}\eta\delta + bq\delta^2 \quad (5.37)$$

with $a_{cc} = 1.507$, $b_{cc} = -0.0984$, $c_{cc} = 0.207$. The parameter \bar{m}_0/M_\odot depends upon the stellar IMF used, but will never be far from unity, and so is ignored. The switch over from pre- to post-cc mass-loss occurs at time $\tilde{t}_{cc} = 16.9\tilde{t}_{rh}^{0.872}$ for Roche lobe filling clusters, and $\tilde{t}_{cc} = 32.0\tilde{t}_{rh}^{0.872}\mathcal{F}_5^{-0.513}$ for Roche lobe under-filling clusters. The numerical values of the various constants in the Roche lobe filling phase are given in Table 5.1, and their values in the Roche lobe under-filling phase are given in Table 5.2.

Although this looks very sophisticated, setting $b = 0$ returns us to the familiar $\frac{dM_c}{dt} \propto -M_c^{1-x} \rho_t^{1/2}$, and since $b \simeq -0.1$ in all cases, these mass-loss rates are in fact only slightly different from those of other authors. Consequently, all these different parameters constitute a very small change from the standard $\mu_{ev} \propto \rho_h^{1/2}$, despite all the added complication. For a cluster initially under-filling its Roche lobe, the expectation is that it will expand as it evolves until it is tidally limited. After this point, it will evolve towards core-collapse. The fraction of mass lost per relaxation time would be expected to change over these processes, and become roughly constant after core-collapse (see section 1.4.4). The Lamers, Baumgardt & Gieles (2010) mass-loss rate attempts to encapsulate all of this physics, hence the multitude of different constants which together combine to make f in the notation of the technical evaporation scalings equation (equation 1.25). Additionally, their mass-loss rate has a built-in expectation about how the mass-loss rate will change with orbital shape, and includes an additional dependence on mass due to the apocentre criterion. Again in the notation of equation (1.25), these are $\tilde{\rho}_t(\tilde{r}_{ev}) = \tilde{\rho}_t(\tilde{r}_p)/(1 - e)$ and $x = \eta$. Therefore these models attempt to account for all of the uncertainties in mass-loss due to evaporation.

As mentioned earlier, Fukushige & Heggie (2000) predicted that the escape time-scale for stars scales with excess energy beyond that required for escape, \hat{E} , as $t_{esc} \propto \hat{E}^{-2}$. By utilising this result, Baumgardt (2001) predicted that under the apocentre criterion the mass-loss rate scales as $\mu_{ev} \propto -M_c^{1-x} \rho_h^{1/2}$. In order to test how sensitive this prediction is to the escape rate energy scaling, Takahashi & Baumgardt (2012) performed a series of Fokker-Plank models with different escape time scalings according to:

$$t_{esc} \propto (\hat{E})^{-\kappa}$$

Takahashi & Baumgardt (2012) modelled $W_0 = 3$ King model cluster evolutions with $\kappa = [1, 2, 3]$, finding that the lifetimes of clusters do not strongly depend on κ , with all lifetimes resembling the $\kappa = 2$ scaling for $N \gtrsim 5 \times 10^4$, and that therefore the $\mu_{ev} \propto -M_c^{1-x} \rho_h^{1/2}$ scaling is robust. They also investigated the effects of internal cluster

structure, corroborating the results of Tanikawa & Fukushige (2005) and Tanikawa & Fukushige (2010) that x depends much more on the strength of the tidal field than on internal cluster structure. As these results were obtained from simulations of clusters on circular orbits, complications from changing tidal radii (i.e. dependence on \tilde{r}_{ev}) were not investigated. Their approximate mass-loss rate can be written in dimensionless notation as:

$$\frac{d\tilde{M}_c}{d\tilde{t}} = - \left(\frac{\nu_e \tilde{M}_c}{2\pi} \right)^{1/(\kappa+2)} \left(\frac{P_0 \tilde{\rho}_t}{3} \right)^{1/2} \left[\left(2 \frac{\tilde{\rho}_h}{\tilde{\rho}_t} \right)^{1/2} \frac{\ln(\lambda \tilde{M}_c)}{0.138} \right]^{\frac{\kappa+1}{\kappa+2}} \quad (5.38)$$

where $\kappa \simeq 2$, and ν_e and λ were determined by comparison with N-body results, and represent the efficiency of mass-loss and the strength of the Coulomb logarithm. They found that $(\lambda, \nu_e) = (0.11, 7)$ works well for both single and multi-mass clusters, and that $(\lambda, \nu_e) = (0.02, 40)$ works well for multi-mass clusters only. Although unclear in their notation, the presence of $\tilde{\rho}_h/\tilde{\rho}_t$ means that their mass-loss rate additionally depends (weakly) on cluster structure. Thus, $x = 1 - 1/(\kappa + 2) = 0.75$ and $f = (\tilde{\rho}_h/\tilde{\rho}_t)^{(\kappa+1)/2(\kappa+2)} = (\tilde{\rho}_h/\tilde{\rho}_t)^{3/8}$.

A large variety of prescriptions by different authors using different methods, making different assumptions, and looking at different aspects of evaporation mass-loss have been brought together. These prescriptions have been summarised in terms of the main uncertainties in evaporation mass-loss rate (cf. equation 1.25) in Table 5.3. The first column displays the equation number in each row, and the second column displays the algebraic and numerical coefficient of each mass-loss rate. The third and fourth columns display the mass and density dependence of each mass-loss rate. The fifth column displays the type of host galaxy generating the tidal field acting on the clusters for which the mass-loss rate was derived, and the final column displays how the mass-loss rate was assumed to depend on orbital shape. The energy criterion mass-loss rates are given in the upper half of the table, with apocentre criterion mass-loss rates in the lower half.

Table 5.3: Breakdown of evaporation mass-loss prescriptions from the literature in terms of coefficient, dependence on mass, cluster structure dependence, assumed host galaxy model, and orbital dependence

Equation	Coefficient	Mass Dependence	Density Dependence	Host Galaxy	Orbit Dependence
(5.12)	$\frac{27}{8} \left(\frac{P_0}{6\pi}\right)^{1/2} \simeq 2.32$	$\ln(0.5\widetilde{M}_c)$	$\tilde{\rho}_t^{-1/2} F(W_0)$	Point Mass	$\tilde{r}_{ev} = \tilde{r}_p$
(5.14)	$\frac{92}{75} P_0^{1/2} \simeq 3.68$	$\ln(\widetilde{M}_{c,i})$	$\tilde{\rho}_t^{-1/2}$	Point Mass	—
(5.15)	$269\xi \ln(\Lambda) \left(\frac{P_0}{4\pi}\right)^{1/2} \simeq 25.46$	—	$\tilde{\rho}_t^{-1/2}$	Isothermal Sphere	$\tilde{r}_{ev} = \tilde{r}_p$
(5.16)	$\tilde{C}_J \left(\frac{P_0}{3}\right)^{1/2} \simeq 10.71$	—	$\tilde{\rho}_h^{-1/2}$		
(5.16)	$\tilde{C}_{MF} \left(\frac{P_0}{3}\right)^{1/2} \simeq 14.02$	—	$\tilde{\rho}_h^{-1/2}$		
(5.16)	$\tilde{C}_{CFM} \left(\frac{P_0}{3}\right)^{1/2} \simeq 7.14$	—	$\tilde{\rho}_h^{-1/2}$		
(5.16)	$\tilde{C}_{Vargo} \left(\frac{P_0}{3}\right)^{1/2} \simeq 10.32$	—	$\tilde{\rho}_h^{-1/2}$		
(5.16)	$\tilde{C}_G \left(\frac{P_0}{3}\right)^{1/2} \simeq 11.15$	—	$\tilde{\rho}_h^{-1/2}$		
(5.17)	$\frac{\xi_0}{0.138} \left(\frac{2P_0}{3\pi}\right)^{1/2} \simeq 0.28$	$\ln(0.4\widetilde{M}_c)$	$\tilde{\rho}_h^{-1/2} \sqrt{1 + \frac{\alpha\tilde{\rho}_t}{2\tilde{\rho}_h}}$	Point Mass	$\tilde{r}_{ev} = \tilde{r}_p$
(5.18)	$\frac{\xi_0}{0.138} \left(\frac{2P_0}{3\pi}\right)^{1/2} \simeq 0.13$	$\ln(0.11\widetilde{M}_c)$	$\tilde{\rho}_h^{-1/2} \exp\left(7.94^3 \sqrt{\frac{\tilde{\rho}_t}{\tilde{\rho}_h}}\right)$	Point Mass	$\tilde{r}_{ev} = \tilde{r}_p$
(5.19)	$100\zeta \left(\frac{P_0}{8\pi\alpha^3}\right)^{1/2} \simeq 216.76$	—	$\tilde{\rho}_t^{-1/2}$	Isothermal Sphere	$\tilde{r}_{ev} = \tilde{r}_M$
(5.20)	$\frac{\xi_0}{0.138^{3/4} k^{1/4}} \left(\frac{2P_0}{3}\right)^{1/2} \simeq 0.06$	$\widetilde{M}_c^{-1/4} \left[\ln(0.11\widetilde{M}_c)\right]^{3/4}$	$\tilde{\rho}_h^{-1/2} \exp\left(7.94^3 \sqrt{\frac{\tilde{\rho}_t}{\tilde{\rho}_h}}\right)$	Point Mass	$\tilde{r}_{ev} = \tilde{r}_p$
(5.21)	$100\zeta \left(\frac{P_0}{8\pi\alpha^3}\right)^{1/2} \simeq 12.19$	$\widetilde{M}_{c,i}^{-1/4} \widetilde{M}_c^{1/4}$	$\tilde{\rho}_t^{-1/2}$	Isothermal Sphere	$\tilde{r}_{ev} = \tilde{r}_M$
(5.29)	$\frac{P_0^{1/2}}{257t_{1000}^{1/2} [1+f(\alpha)q(\tilde{r}_t)]} 1.5 \times 10^{5x(\tilde{r}_t)} \simeq 0.015$	$\widetilde{M}_{c,i}^{1-x(\tilde{r}_t)}$	$\tilde{\rho}_{h,i}^{-1/2}, \tilde{r}_t = \frac{2\tilde{\rho}_{h,i}}{\tilde{\rho}_{t,i}}$	Power-laws, $k \in [1, 3]$	—
(5.33)	$\left(\frac{340}{9t_{ref}^N}\right)^q a^{q\eta} \left(\frac{P_0}{6}\right)^{q/2} \simeq 19.51, 7.61$	$\widetilde{M}_{c,i}^{bq\eta\delta} \widetilde{M}_c^{1-\eta}$	$\tilde{\rho}_t^{-q/2}$	Isothermal Sphere	$(1-e)^{-q}$
(5.35)	$\left(\frac{340}{9t_{ref}^N}\right)^q \frac{a_{cc}^7 a \rho_0^{\delta}}{6.48 A F_t^{-0.127+c_{cc}+q\delta}} \left(\frac{P_0}{6}\right)^{q/2} \simeq 5.52, 5.29$	$\widetilde{M}_{c,i} \widetilde{M}_c^{1-\eta}$	$\tilde{\rho}_t^{-q/2}$	Isothermal Sphere	$(1-e)^{-q}$
(5.38)	$\left(\frac{\mu_c}{2\pi}\right)^q \frac{1}{\kappa+2} \left(\frac{\sqrt{2}}{0.138}\right)^{\frac{\kappa+1}{\kappa+2}} \left(\frac{P_0}{3}\right)^{1/2} \simeq 10.19, 15.76$	$\widetilde{M}_c^{\frac{1}{\kappa+2}} \left[\ln(\lambda\widetilde{M}_c)\right]^{\frac{\kappa+1}{\kappa+2}}$	$\tilde{\rho}_t^{-1/2} \left(\frac{\tilde{\rho}_t}{\tilde{\rho}_h}\right)^{\frac{\kappa+1}{2(\kappa+2)}}$	—	—

The numerical coefficients in the second column were calculated assuming a value of $P_0 = 9$, with algebraic constants given in the discussion of the relevant mass-loss rate. For the coefficients of the mass-loss rate obtained via reduced χ^2 minimisation (equation 5.16), an average stellar mass of $\bar{m} = 0.7 M_\odot$ was assumed. For the evaluation of the numerical coefficient of the Tanikawa & Fukushige (2010) mass-loss rate (equation 5.29), the host galaxy was assumed to be an isothermal sphere ($\alpha = 2$), and a value of $x(\hat{r}_t) = 0.75 \rightarrow \hat{r}_t = 4.16$ was assumed. The numerical coefficients of the Lamers, Baumgardt & Gieles (2010) mass-loss rates were calculated with the constants for a $\widetilde{M}_c > 10^3$, $W_0 = 7$ cluster in both pre-cc and post-cc phases for Roche lobe filling clusters (equation 5.33, cf. Table 5.1). For the Roche lobe under-filling mass-loss rates (equation 5.35), the constants for a $\widetilde{M}_c > 10^3$, $\mathcal{F}_7 = 0.4$ cluster were assumed for the calculation of both the pre-cc and post-cc mass-loss rate coefficients (cf. Table 5.2). In both cases, the first coefficient is for pre-cc, and the second is for post-cc. Finally, two numerical coefficients for the Takahashi & Baumgardt (2012) mass-loss rate (equation 5.38) were calculated using the two values of $\nu_e = 7, 40$ presented by the authors, and assuming $\kappa = 2$.

The numerical evaluation of coefficients in the second column of Table 5.3 have rather a large spread in values, from 0.06 to 216.76. Of course, the rest of the evaporation mass-loss equation must also be considered before declaring that the mass-loss caused by one prescription will be more severe than another. For example, assuming a cluster mass of $\widetilde{M}_c = 10^5$ and half-mass density $\tilde{\rho}_h = 150\tilde{\rho}_t \Rightarrow \tilde{r}_h \simeq 0.15\tilde{r}_t$ would make the mass-loss rate of equation (5.20) scale as $d\widetilde{M}_c/d\tilde{t} \simeq 310\tilde{\rho}_t^{1/2}$, while the mass-loss rates of equation (5.16) would scale as $d\widetilde{M}_c/d\tilde{t} \simeq 130\tilde{\rho}_t^{1/2}$. On the other hand, the numerical coefficient on equation (5.15) is exactly what is displayed for the same conditions, and therefore predicts mass-loss rates a factor of several smaller. Likewise, the coefficient on equation (5.29) would predict a mass-loss rate of $d\widetilde{M}_c/d\tilde{t} \simeq 3.27\tilde{\rho}_t^{1/2}$. However this mass-loss rate is based on the initial conditions of the cluster, and therefore where the mass-loss of other prescription will decelerate as cluster mass becomes very small, this prescription will continue unabated. Consequently it would be natural to expect a smaller coefficient in order to achieve a similar overall mass-loss. However,

the mass-loss rate of equation (5.29) with the example mass and half-mass density given above has a coefficient of around 5000. Obviously this is extremely high and would require a CIMF extending up to very high masses to have any clusters left after 13 Gyr of evolution. The reason this mass-loss rate is so high is likely because this evaporation prescription was presented by Takahashi & Baumgardt (2012) qualitatively rather than quantitatively, and so could well be missing some numerical factors (e.g. a factor of $\widetilde{M}_{c,i}^{-1/(\kappa+2)}$ would likely remedy the anomalously large coefficient). Therefore under the same conditions, these different evaporation prescriptions can be expected to produce a range of mass-loss rates, those with typically high mass-loss rates will destroy many clusters, while those with low mass-loss rates will destroy fewer. While investigating the number of clusters destroyed during the course of GCMF evolution has consequences for what fraction of stars present in the Galaxy today may have originated in clusters, the focus remains on the shape of the resulting GCMF rather than its scale.

The specific details of how these evaporation prescriptions will affect GCMF shape evolution depends on how they scale with cluster properties. For example, energy criterion prescriptions with variable Coulomb logarithms will become slightly weaker as clusters lose a significant fraction of their mass, while prescriptions under the apocentre criterion will become much weaker as clusters lose a significant fraction of their mass. Similarly, prescriptions depending on cluster structure are weaker when acting on clusters with smaller $\tilde{\rho}_t/\tilde{\rho}_h$ at fixed $\tilde{\rho}_h$, i.e. clusters that are more concentrated and/or significantly under-filling their Roche lobes, as expected based on the discussion in section 1.4.4. Therefore, when employing these evaporation prescriptions, those that are weaker for more concentrated clusters would be expected to produce GCSs with a higher prevalence of more concentrated clusters than prescriptions that do not depend on cluster structure. Likewise, prescriptions that are weaker when acting on lower mass clusters would be expected to produce GCSs with a higher prevalence of low-mass clusters than prescriptions that do not depend on cluster mass. Furthermore, prescriptions with effective tides acting from orbital pericentre will affect otherwise similar clusters on elliptical orbits much more strongly, and consequently an evolved GCS favouring less elliptical orbits would be expected, and similarly for prescriptions

with an inverse dependence on orbital ellipticity. Prescriptions assuming $\tilde{r}_{ev} = \tilde{r}_M$, (i.e. effective tides acting from the current galactocentric radius) is essentially equivalent to effective tides acting from the radius where $\tilde{\rho}_t^{1/2}(\tilde{r}_{ev}) = \bar{\rho}_t^{1/2}$, where a bar denotes a time average. Exactly where this radius will lie will depend on the host galaxy, but is likely to be near the orbital apocentre since this is where clusters move slowest and consequently spend most of their time. On the other hand, prescriptions with the same dependences but different coefficients will end up with the same proportion of low/high mass/density/orbital ellipticity clusters, with the only difference being in the number of survivors after a Hubble time of evolution, i.e. a stronger evaporation prescription will result in a less populous evolved GCS.

In terms of radial dependence of the resulting GCMF evolved with each of these prescriptions, there are four possible scenarios to consider — a cluster population will have initial properties that either depend on initial galactocentric distance or do not, and subsequent evaporation that either depends on galactocentric distance or does not. For example, Elson, Fall & Freeman (1987) conducted a study of Young Massive Clusters (YMCs) in the LMC, finding that most if not all have unbound halos of stars. Given the young age of these clusters ($\lesssim 1\text{Gyr}$), Elson, Fall & Freeman (1987) argue that the origin of these halos of unbound stars is tidal stripping, and that the LMC YMCs were born filling their Roche lobes. On the other hand, authors such as Baumgardt et al. (2010) and Gieles et al. (2011) contend that a significant fraction of even Milky Way GCs are not yet tidally limited on the basis of the ratio of half-mass radii to tidal radii corresponding to their current galactocentric radii, and models assuming an infinite initial density and subsequent expansion to fill their Roche lobes. Thus, there are arguments both ways as to whether initial half-mass densities and tidal densities, or equivalently half-mass densities and galactocentric radii are correlated in some way. Subsequently, an evolving GCS may have further radial dependence built in by evaporation prescriptions with a strong radial dependence (e.g. $\tilde{r}_{ev} = \tilde{r}_p$, or $\mu_{ev} \propto \tilde{\rho}_t^{1/2}$, cf. equation 1.8). Alternatively, a GCS evolved with prescriptions that have no radial dependence (e.g. $\mu_{ev} \propto \tilde{\rho}_h^{1/2}$) will only continue to have whatever radial dependence they were initialised with. For scenarios with a radial dependence (either

primordial, through the prescription, or both), the distribution of cluster orbits will play a role in determining the final radial dependence of the GCS, and of particular interest is the GCMF.

5.2.2 Tidal Shocks

Gnedin & Ostriker (1997) modelled the dynamical evolution of the Milky Way globular cluster system including effects of evaporation and tidal shocks. For evaporation, they adopted a very simple form, $dM_c/dt = \xi M_c/t_{rh}$, with $\xi = 0.045$ (Hénon 1961) and t_{rh} the standard relaxation time (Spitzer 1987, equation 1.4). They used a far more detailed description of tidal shocks, including a shock induced relaxation term (the second order term in the average energy change for stars in a tidally shocked cluster; Kundic & Ostriker (1995), see also section 1.4.3). These first and second order average energy change terms for stars at cluster-centric radius \tilde{r} with orbital frequencies $\tilde{\Omega}$ are:

$$\langle \Delta \tilde{E} \rangle_{bulge} = \frac{4}{3} \left(\frac{\tilde{M}_b}{\tilde{v}_p \tilde{r}_p^2} \right)^2 \tilde{r}^2 A_b(x_b) \chi(\tilde{r}_p) \lambda(\tilde{r}_p, \tilde{r}_a) \quad (5.39)$$

$$\langle \Delta \tilde{E}^2 \rangle_{bulge} = \frac{8}{9} \left(\frac{\tilde{M}_b}{\tilde{v}_p \tilde{r}_p^2} \right)^2 \tilde{\Omega}^2 \tilde{r}^4 A_b(x_b) \chi(\tilde{r}_p) \lambda(\tilde{r}_p, \tilde{r}_a) \quad (5.40)$$

where \tilde{M}_b is the mass causing the tidal shock, \tilde{r}_p is the orbital pericentre, \tilde{v}_p is pericentric cluster speed, and $A_b(x_b)$, $\chi(\tilde{r}_p)$ and $\lambda(\tilde{r}_p, \tilde{r}_a)$ are correction functions described below.

The periods of stars in the outer regions of a cluster will typically be greater than the duration of a gravitational encounter during a cluster pericentric passage. Thus constituent stars are frequently assumed to be stationary, known as the impulse approximation. However, stars in the central regions of a cluster move much faster and may react to shocks adiabatically, and consequently the impulse approximation breaks down.

To correct for this, an adiabatic correction factor was introduced (Weinberg 1994a,b,c):

$$A_b(x_b) = (1 + x_b)^{-3/2} \quad (5.41)$$

$$x_b = \frac{P_0 \tilde{\rho} \tilde{r}_p^2}{3 \tilde{v}_p^2} \quad (5.42)$$

where $\tilde{\rho}(\tilde{r})$ is the average density of the cluster inside of radius \tilde{r} . Since the average density is related to circular angular frequency and circular orbital period by $\tilde{\rho} \propto \tilde{\Omega}^2 \propto \tilde{T}^{-2}$, x_b is essentially the ratio of the period of a circular orbit at radius \tilde{r} in the cluster to the duration of the shock. Thus when the duration of the encounter is much less than a typical orbital period, $x_b \simeq 0$ and $A_b \simeq 1$, whereas when the ratio of shock duration to typical orbital period is greater, $x_b \gg 0$ and $A_b \ll 1$. Consequently, equations (5.39) and (5.40) are greatly reduced, reflecting the adiabatic conditions for the star. The best typical representation of the orbital periods of stars will be that of a star at the half-mass radius, and thus the affect of the shock is calculated assuming that the local density in the cluster is the half-mass density, $\tilde{\rho} = \tilde{\rho}_h$.

Furthermore, a correction accounting for the distended nature of the mass distribution producing the shock (rather than simply assume it is a point mass; Gnedin, Hernquist & Ostriker 1999) was introduced:

$$\chi(\tilde{r}_p) = 0.5 [(3J_0 - J_1 - I_0)^2 + (2I_0 - I_1 - 3J_0 + J_1)^2 + I_0^2] \quad (5.43)$$

$$I_0(\tilde{r}_p) = \int_1^\infty \frac{\tilde{M}(\zeta \tilde{r}_p)}{\tilde{M}_b} \frac{d\zeta}{\zeta^2(\zeta^2 - 1)^{1/2}} \quad (5.44)$$

$$I_1(\tilde{r}_p) = \int_1^\infty \frac{\tilde{r}_p}{\tilde{M}_b} \frac{d\tilde{M}}{d\tilde{r}}(\zeta \tilde{r}_p) \frac{d\zeta}{\zeta^2(\zeta^2 - 1)^{1/2}} \quad (5.45)$$

$$J_0(\tilde{r}_p) = \int_1^\infty \frac{\tilde{M}(\zeta \tilde{r}_p)}{\tilde{M}_b} \frac{d\zeta}{\zeta^4(\zeta^2 - 1)^{1/2}} \quad (5.46)$$

$$J_1(\tilde{r}_p) = \int_1^\infty \frac{\tilde{r}_p}{\tilde{M}_b} \frac{d\tilde{M}}{d\tilde{r}}(\zeta \tilde{r}_p) \frac{d\zeta}{\zeta^4(\zeta^2 - 1)^{1/2}} \quad (5.47)$$

For a point mass system, $I_0 = 1$, $J_0 = 2/3$, and $I_1 = J_1 = 0$, and thus $\chi = 1$ everywhere, whereas for distended mass distributions, $\chi(\tilde{r}_p) \geq 1$ everywhere. These integrals are displayed for the Hernquist (1990) model in Figure 5.1.

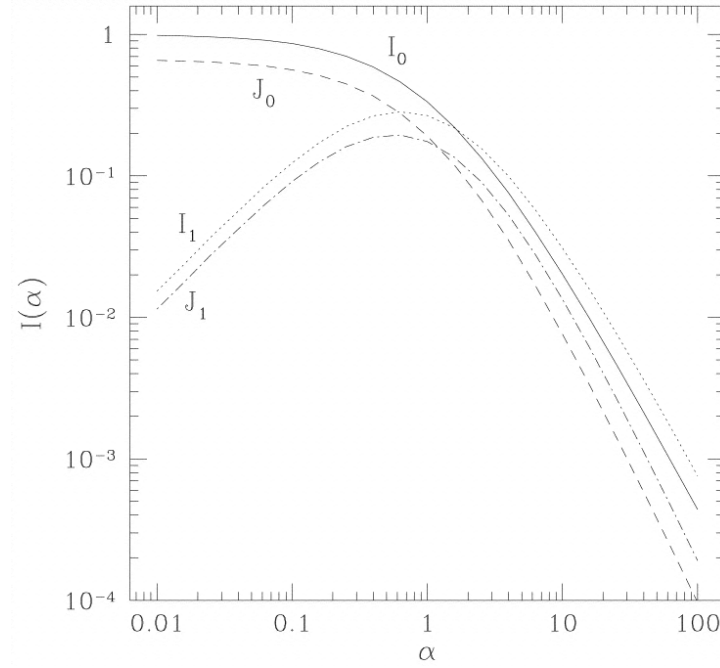


Figure 5.1: Point mass correction integrals for the Hernquist (1990) model. Figure taken from Gnedin, Hernquist & Ostriker (1999).

Finally, a correction accounting for time variation in the tidal field strength between pericentre and apocentre (Aguilar, Hut & Ostriker 1988) was also included:

$$\lambda(\tilde{r}_p, \tilde{r}_a) = \left[\frac{\widetilde{M}_b(\tilde{r}_a)}{\widetilde{M}_b(\tilde{r}_p)} \left(\frac{\tilde{r}_p}{\tilde{r}_a} \right)^3 - 1 \right]^2 \quad (5.48)$$

For circular orbits, $\tilde{r}_p = \tilde{r}_a$, and $\lambda(\tilde{r}_p, \tilde{r}_a) = 0$, making $\langle \Delta \tilde{E} \rangle_{bulge} = \langle \Delta \tilde{E}^2 \rangle_{bulge} = 0$ (i.e. no tidal shock), as required. With all the necessary ingredients included, the shocking time-scale is then given by:

$$\tilde{t}_{shock} = \tilde{T}_r \frac{|\tilde{E}_{bind}|}{\langle \Delta \tilde{E} \rangle_{bulge}} + \tilde{T}_r \frac{|\tilde{E}_{bind}|^2}{\langle \Delta \tilde{E}^2 \rangle_{bulge}} \quad (5.49)$$

where Gnedin & Ostriker (1997) assumed $|\tilde{E}_{bind}| \simeq 0.2 \widetilde{M}_c / \tilde{r}_h$ is the total cluster binding energy, and \tilde{T}_r is the cluster radial period. Then the mass-loss rate is given by

$$d\widetilde{M}_c/d\tilde{t} = -\widetilde{M}_c/\tilde{t}_{shock}:$$

$$\frac{d\widetilde{M}_c}{d\tilde{t}} = -\frac{80}{3P_0} \left(\frac{\widetilde{M}_b}{\tilde{v}_p \tilde{r}_p^2} \right)^2 \frac{\widetilde{M}_c A_b(x_b) \chi(\tilde{r}_p) \lambda(\tilde{r}_p, \tilde{r}_a)}{\tilde{T}_r \tilde{\rho}_h} \quad (5.50)$$

where the “bulge” is taken to be the portion of the host galaxy interior to the cluster pericentre. Inspection of equation (5.50) reveals the kinds of scalings expected based on the arguments in section 1.4.3 — i.e. in terms of cluster parameters, the rate of mass-loss due to tidal shocking scales as $\mu_{sh} \propto \widetilde{M}_c/\tilde{\rho}_h$, while in terms of orbital and host galaxy dependence, the mass-loss rate scales as $\mu_{ev} \propto \widetilde{M}_b^2/\tilde{v}_p^2 \tilde{r}_p^4 \tilde{T}_r$. The effect of a very adiabatic shock (the limit of $P_0 \tilde{\rho}_h \tilde{r}_p^2/3\tilde{v}_p^2 \gg 1$) is to change these scalings to $\mu_{ev} \propto \widetilde{M}_c/\tilde{\rho}_h^{5/2}$ in terms of cluster properties, and $\mu_{ev} \propto \widetilde{M}_b^2 \tilde{v}_p/\tilde{r}_p^7 \tilde{T}_r$ in terms of orbital and host galaxy dependence. The point mass and time-varying tidal field corrections do not affect how the shocking mass-loss rate scales with cluster properties, but make the shocking mass-loss rate more rapid for more elliptical orbits and more distended host galaxies. Therefore, tidal shocking would be expected to deplete a GCS of high-mass, low-density GCs on short-period elliptical orbits.

5.2.3 Stellar Evolution

The mass-loss rate due to stellar evolution can be calculated by taking the time derivative of equation (1.14), the initial mass minus the mass lost due to stellar evolution at time t (see section 1.4.1). This gives:

$$\frac{d\widetilde{M}_c}{d\tilde{t}} = \frac{dN}{d \log m} \bigg|_{m_{to}} \frac{d \log m_{to}}{d \log \tilde{t}} \left[\frac{m_{to} - \{1 - P_{ej}(m_{to})\} m_{rm}(m_{to})}{\tilde{t} \ln(10)} \right] \quad (5.51)$$

where m_{to} is the mass of a star evolving off of the main-sequence at time \tilde{t} , $P_{ej}(m)$ is the probability that the remnant of a star of progenitor mass m will be ejected by its kick velocity as it expires, $m_{rm}(m)$ is the mass of the remnant left after a star of progenitor mass m expires, and $dN/d \log m$ is the logarithmic initial mass function defined between the most and least massive stars allowed, m_u and m_l . This treatment

assumes that a star will instantly turn into a remnant once it evolves off of the main-sequence, and ignores mass-loss due to stellar winds while on the main-sequence. By allowing stars to evolve, expire, and possibly escape, the average stellar mass in a cluster must change with time to be self-consistent. Ignoring all other forms of mass-loss, the number of particles (stars and remnants) in a cluster can only be changed by remnants escaping due to an imparted kick velocity. Therefore, upon the expiry of stars of mass m , the cluster will lose dN_{loss} particles:

$$dN_{loss}(m) = \frac{dN}{d \log m} P_{ej}(m) d \log m \quad (5.52)$$

Then the remaining fraction of particles at time \tilde{t} is given by:

$$\frac{N_{stev}(\tilde{t})}{N_{init}} = 1 - \frac{1}{N_{init}} \int_{\log m_l}^{\log m_u} \frac{dN}{d \log m} P_{ej}(m) d \log m \quad (5.53)$$

with

$$N_{init} = \int_{\log m_l}^{\log m_u} \frac{dN}{d \log m} d \log m \quad (5.54)$$

Then the average stellar mass, \bar{m} at time \tilde{t} can be calculated by taking the ratio of $M_{stev}(\tilde{t})/M_{init}$ (equations 1.14 and 1.15) divided by equation (5.53):

$$\bar{m}(\tilde{t}) = \frac{1 - \frac{1}{M_{init}} \int_{\log m_{to}}^{\log m_u} \frac{dN}{d \log m} [m - \{1 - P_{ej}(m)\} m_{rm}(m)] d \log m}{1 - \frac{1}{N_{init}} \int_{\log m_{to}}^{\log m_u} \frac{dN}{d \log m} P_{ej}(m) d \log m} \quad (5.55)$$

Furthermore, taking the time derivative of \bar{m} yields:

$$\frac{d\bar{m}}{d\tilde{t}} = \frac{\bar{m}}{\tilde{t} \ln(10)} \frac{dN}{d \log m} \bigg|_{m_{to}} \frac{d \log m_{to}}{d \log \tilde{t}} \left[\frac{m_{to} - \{1 - P_{ej}(m_{to})\} m_{rm}(m_{to})}{M_c} - \frac{P_{ej}(m_{to})}{N_{stev}} \right] \quad (5.56)$$

In order to calculate these, m_{to} as a function of \tilde{t} is required. This is provided by Chernoff & Weinberg (1990) (hereafter referred to as CW) in a table reporting on the results of Miller & Scalo (1979) for Population I stars with main-sequence lifetimes in $[10^{0.5}, 10^{4.18}]$ Myr, or a more modern version is provided by Hurley, Pols & Tout (2000) (hereafter referred to as HPT). At times before $m_{to}(\tilde{t}) = m_u$, no stars present will have expired yet, and thus there is no mass-loss due to stellar evolution.

The different stellar evolution prescriptions pertain to the choice of $m_{to}(\tilde{t})$ as described by CW or HPT, along with their different choices for remnant type as a function of progenitor mass. The remnant mass as a function of progenitor mass as used by CW is:

$$m_{rm}(m) = \begin{cases} 0.58 + 0.22(m - 1) & m < 4.7 \\ 0 & 4.7 \leq m \leq 8 \\ 1.4 & m > 8 \end{cases} \quad (5.57)$$

where the descending progenitor mass inequalities correspond to white dwarfs, an explosion sufficiently violent such that no remnant is left, and neutron stars (Iben & Renzini 1983) (with m in units of M_\odot). The remnant mass as a function of progenitor mass given by HPT is defined in terms of the core mass at the base of the asymptotic giant branch, $M_{AG} = (0.000436(m)^{5.22} + 0.0684)^{0.25}$. The mass ranges for different classes of remnant in this case are:

$$m_{rm}(m) = \begin{cases} M_{AG} & m \leq 3.58 \\ 0.44M_{AG} + 0.448 & 3.58 < m \leq 8.19 \\ 1.17 + 0.09 \max(1.44, 0.77M_{AG} - 0.35) & m > 8.19 \end{cases} \quad (5.58)$$

where m , m_{rm} , and M_{AG} are in units of M_\odot , and the descending inequalities correspond to white dwarfs that have a second dredge-up prior to expiry, white dwarfs that do not have a second dredge-up prior to expiry, and progenitors massive enough to become either neutron stars or black holes. Thus, any progenitor with $M_{AG} \leq 2.25 M_\odot \Rightarrow m \leq 8.19 M_\odot$ will end up as a white dwarf. Additionally, HPT assume a black hole remnant for any star with $M_{AG} > 9.52 M_\odot \Rightarrow m > 24.76 M_\odot$.

In order to obtain the time derivative of $m_{to}(\tilde{t})$, a function is fit to the data points, and its derivative taken. Figure 5.2 displays m_{to} as a function of \tilde{t} as calculated by Miller & Scalo (1979), the functional fit to these data, and the functional fit to the HPT m_{to} as a function of \tilde{t} data as derived by Lamers, Baumgardt & Gieles (2010) for lowest metallicity stars. It is apparent that the functional fit for the Miller & Scalo (1979) model is of questionable accuracy for the first few Myr. However the evolution of the GCS is begun with the assumption that the extreme mass-loss caused by the expiry of very massive stars at these early times has already transpired, and these fits are only used for times after this period ($\tilde{t}_{start} = 30$).

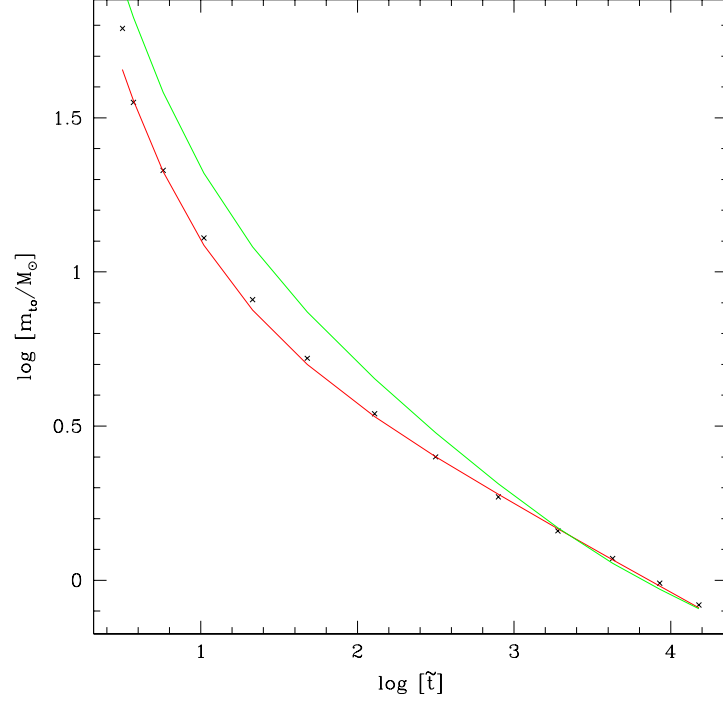


Figure 5.2: Progenitor mass of stars evolving off of the main-sequence as a function of time, $m_{to}(\tilde{t})$. Black points are Miller & Scalo (1979) model data as presented in Chernoff & Weinberg (1990). The red curve is the functional fit to these data, and the green curve is the Lamers, Baumgardt & Gieles (2010) functional fit to the lowest metallicity Hurley, Pols & Tout (2000) model, the data points of which are absent.

The derivatives $dm_{to}/d\tilde{t}$ of these models are then taken to be the derivatives of the functional fits in Figure 5.2:

$$\frac{d \log m_{to}}{d \log \tilde{t}} = -0.7735(\log \tilde{t})^{0.3} \exp [0.7 (11.1695 - (\log \tilde{t} + 6)^{1.3})] - 0.28 \quad (5.59)$$

for the Miller & Scalo (1979) models, and

$$\frac{d \log m_{to}}{d \log \tilde{t}} = -0.3864 + 0.11256[\log \tilde{t}] + 0.04572[\log \tilde{t}]^2 - 0.02951[\log \tilde{t}]^4 \quad (5.60)$$

for the Hurley, Pols & Tout (2000) models.

During GCS evolution runs involving stellar evolution, the stellar IMF is always taken to be that of a Chabrier (2003) disk IMF, and the probability of escape upon

becoming a remnant, P_{ej} , can be obtained from theoretical estimates, (e.g. Heyl 2007; Woosley & Heger 2007). Ultimately P_{ej} is determined by a user input value for each remnant class (i.e. white dwarfs, neutron stars, or black holes). Assuming that other mass-loss mechanisms that may be acting on the cluster do not preferentially target stars of a particular mass, then the number of particles present at time \tilde{t} is simply $N(\tilde{t}) = \widetilde{M}_c(\tilde{t})/\bar{m}$. However, this is not accurate for example for mass segregated clusters undergoing evaporation, as these clusters preferentially lose low-mass stars — however clusters may take a significant amount of time to achieve mass segregation, so this assumption should be reasonable for the first few relaxation times.

Therefore, these two stellar evolution prescriptions only differ in two ways; the remnant mass function, $m_{rm}(m)$, and $m_{to}(\tilde{t})$, the function describing the mass of a star evolving off of the main-sequence at time \tilde{t} . The more massive remnant a particular mass of progenitor evolves into (with the same non-zero probability of being retained), the less mass a host cluster will lose. The remnant mass predicted by HPW and CW for a progenitor mass of $m = 2 M_\odot$ is the same, at $m_{rm} = 0.8 M_\odot$, with the CW prediction smaller at lower mass progenitors, and larger at higher mass progenitors. Between the progenitor masses of $[4.7, 8] M_\odot$, HPT predict remnants of masses between $[0.93, 1.41] M_\odot$, whereas CW predict no remnants at all. At progenitor masses just over $m = 8 M_\odot$, both remnant mass predictions are basically the same, however the CW remnant mass remains constant at $m_{rm} = 1.4 M_\odot$, whereas the HPT remnant continues to slowly increase (e.g. a progenitor mass of $24.76 M_\odot$ is predicted to become a remnant black hole of mass $1.8 M_\odot$). Thus, given that in a Chabrier (2003) disk IMF, most of the total mass is in low-mass stars, and given the absence of any remnant for intermediate masses in the CW prescription, it seems likely that a cluster with stellar evolution according to the HPT prescription would retain more mass in remnants. Additionally, examination of equation (5.51) reveals that there is no dependence on cluster properties or environmental parameters. Therefore, since the remnant mass as a function of progenitor mass, probability of ejection as a function of progenitor mass, main-sequence turn-off mass as a function of time, and the shape of the stellar IMF are assumed to be universal, the only detail which varies from one cluster to another is

the normalisation of the stellar IMF (equation 1.15). Consequently every member of a GCS will lose the same fraction of its initial mass by time \tilde{t} , and therefore the evolved GCMF will only be shifted to lower masses, with the overall shape remaining constant (modulo clusters that end up at masses low enough to be considered destroyed).

In terms of the mass of stars evolving off of the main-sequence at time \tilde{t} , HPT predict a function $m_{to}(\tilde{t})$ that is at higher mass for the first Gyr or so, before matching the function used by CW at later times (cf. Figure 5.2). Therefore at the same \tilde{t} , HPT predict greater $d \log m_{to} / d \log \tilde{t}$ and m_{to} , and consequently a greater rate of mass-loss (cf. equation 5.51). Therefore in conclusion, compared to the Chernoff & Weinberg (1990) stellar mass-loss prescription, the Hurley, Pols & Tout (2000) prescription will likely result in clusters with lower mass overall, but with a greater percentage of cluster mass in remnants.

Stellar evolution is not expected to be of first order importance with regards to shaping the GCMF over a Hubble time of evolution, due to the fact that stellar evolution removes the same fraction of mass from every GC, leaving the shape of the GCMF unchanged (e.g. see section 1.4.1). However, stellar evolution can remove a significant fraction of mass from all GCs, and this may in turn affect how other destruction mechanisms (i.e. tidal shocking and dynamical friction) act on them. Therefore in the interests of allowing investigation into the interplay between destruction mechanisms, this simple treatment is employed when stellar evolution is selected as a destruction mechanism.

5.2.4 Dynamical Friction

Under the assumption that the velocity distribution of the host galaxy is everywhere identical, that the host galaxy is infinite and homogeneous, and that the body experiencing dynamical friction is much more massive than the particles the host galaxy consists of, Chandrasekhar (1943) calculated the deceleration of the subject body due

to dynamical friction as (e.g. Binney & Tremaine 2008):

$$\tilde{a}_{df} = -P_0 \left(\frac{\bar{m}_0}{M_0} \right) \frac{\tilde{\rho} \tilde{M}_c \ln \Lambda}{\tilde{v}_M^2} \left[\text{erf}(X) - \frac{2X}{\sqrt{\pi}} e^{-X^2} \right] \quad (5.61)$$

$$X = \frac{\tilde{v}_M}{\sqrt{2}\tilde{\sigma}} \quad (5.62)$$

where \bar{m}_0/M_0 is the ratio of a typical tracer's initial average stellar mass to the mass-scale of the galaxy, $\tilde{\rho}$ is the local galaxy density, $\ln \Lambda$ is the Coulomb logarithm, and $\tilde{\sigma}$ is the local one dimensional velocity dispersion of the background galaxy, given by equation (3.43). Despite the limiting assumptions about the host galaxy velocity distribution and structure, under extensive study Chandrasekhar's dynamical friction formula has been found to give remarkably good results when compared with self-consistent N-body simulations (e.g. White 1983; Bontekoe & van Albada 1987; Zaritsky & White 1988; Cora, Vergne & Muzzio 1997; Cora, Vergne & Muzzio 2001; Fabio & Merritt 2012). However, there are some areas where Chandrasekhar's dynamical friction formula does not fare so well, for example in constant density cores where the dynamical friction force is suppressed and can actually disappear entirely (Read et al. 2006a; Inoue 2009). This can easily be accounted for by simply switching dynamical friction off when a tracer enters a constant density core.

By fitting for free parameters in the Coulomb logarithm using N-body results, Just & Peñarrubia (2005) derive the following expression for the Coulomb logarithm:

$$\ln \Lambda = \ln \frac{\tilde{b}_{max}}{\sqrt{\tilde{b}_{min}^2 + \tilde{a}_{90}^2}} \quad (5.63)$$

$$\tilde{b}_{max} = \min \left(\tilde{r}_M, \frac{\tilde{\rho}}{d\tilde{\rho}/d\tilde{r}} \right) \quad (5.64)$$

$$\tilde{b}_{min} = \tilde{r}_h \quad (5.65)$$

$$\tilde{a}_{90} = \frac{\tilde{M}_c}{2\tilde{\sigma}^2 + \tilde{v}_M^2} \frac{\bar{m}_0}{M_0} \quad (5.66)$$

where \tilde{b}_{max} and \tilde{b}_{min} are the maximum and minimum impact parameters (see section 1.4.2), $\tilde{\sigma}$ is the local host system velocity dispersion (obtained through solving equation (3.42), the spherical isotropic Jeans equation), and \tilde{a}_{90} is the typical impact parameter

for a 90 degree deflection. Using this result plus self consistent velocity distributions (made possible through the use of scale free models), Just et al. (2011) achieve orbital decay times an order of magnitude in better agreement with N-body simulations than when using a constant Coulomb logarithm.

5.2.4.1 Ellipticity of Decaying Orbits

The quantity $d\tilde{L}_c(\tilde{E})/d\tilde{L}$ can be used to indicate whether dynamical friction is placing the tracer in question on a more or less elliptical orbit, as follows. Recalling that $h = \tilde{L}/[\tilde{L}_c(\tilde{E}) + \tilde{L}_0]$,

$$\frac{dh}{d\tilde{t}} = \frac{1}{\tilde{L}_c(\tilde{E}) + \tilde{L}_0} \frac{d\tilde{L}}{d\tilde{t}} \left[1 - \frac{\tilde{L}}{\tilde{L}_c(\tilde{E}) + \tilde{L}_0} \frac{d\tilde{L}_c(\tilde{E})}{d\tilde{L}} \right] \quad (5.67)$$

Since $\tilde{L}_c(\tilde{E}) \geq 0$, $\tilde{L} \geq 0$, $\tilde{L}_0 \geq 0$ and $d\tilde{L}/d\tilde{t} \leq 0$, this means that:

$$\begin{aligned} h \frac{d\tilde{L}_c(\tilde{E})}{d\tilde{L}} < 1 &\Rightarrow \frac{dh}{d\tilde{t}} < 0 \equiv \text{increasing ellipticity} \\ h \frac{d\tilde{L}_c(\tilde{E})}{d\tilde{L}} = 1 &\Rightarrow \frac{dh}{d\tilde{t}} = 0 \equiv \text{constant ellipticity} \\ h \frac{d\tilde{L}_c(\tilde{E})}{d\tilde{L}} > 1 &\Rightarrow \frac{dh}{d\tilde{t}} > 0 \equiv \text{decreasing ellipticity} \end{aligned}$$

The quantity $h d\tilde{L}_c(\tilde{E})/d\tilde{L}$ may be obtained through repeated application of the chain rule, to give:

$$h \frac{d\tilde{L}_c(\tilde{E})}{d\tilde{L}} = h \frac{d\tilde{L}_c}{d\tilde{r}_E} \frac{d\tilde{r}_E}{d\tilde{E}} \frac{d\tilde{E}}{d\tilde{t}} \frac{d\tilde{t}}{d\tilde{L}} \quad (5.68)$$

where \tilde{r}_E is the radius of a circular orbit with energy \tilde{E} ; i.e. such that:

$$\tilde{E} = \tilde{\Phi}(\tilde{r}_E) + \frac{\tilde{r}_E}{2} \frac{d\tilde{\Phi}}{d\tilde{r}} \bigg|_{\tilde{r}_E} \quad (5.69)$$

$$\tilde{L}_c = \sqrt{\tilde{r}_E^3 \frac{d\tilde{\Phi}}{d\tilde{r}} \bigg|_{\tilde{r}_E}} \quad (5.70)$$

Then with equations (5.7) and (5.8) it can finally be shown that:

$$h \frac{d\tilde{L}_c(\tilde{E})}{d\tilde{L}} = \frac{h\tilde{v}_M^2}{\tilde{L}\tilde{\Omega}(\tilde{E})} \quad (5.71)$$

where $\tilde{\Omega}(\tilde{E})$ is the angular frequency of a circular orbit with energy \tilde{E} . For an initially circular orbit, $\tilde{L} = \tilde{v}_M^2/\tilde{\Omega}(\tilde{E})$, and thus $d\tilde{L}_c(\tilde{E})/d\tilde{L} = 1$. If $\tilde{L}_0 = 0$, then $h = 1$ for a circular orbit, and this is another expression of the well known result that a subject body initially on a circular orbit will migrate through a sequence of circular orbits as its orbital energy and angular momentum decay, (e.g. van den Bosch et al. 1999; Binney & Tremaine 2008). However, if $\tilde{L}_0 > 0$, then $h = \tilde{L}/[\tilde{L}_c(\tilde{E}) + \tilde{L}_0] < 1$ and consequently $h d\tilde{L}_c(\tilde{E})/d\tilde{L} < 1$ for an initially circular orbit. Thus the subject body must migrate to a higher ellipticity orbit as its orbital energy and angular momentum decay. Moreover, equation (5.71) indicates that this is true regardless of the form of \tilde{a}_{df} . Note that the quantity in equation (5.71) only indicates whether a subject body is migrating to a more or less elliptical orbit at a single instant in time. In order to deduce whether a subject body will end up on a more or less elliptical orbit after a period of time would require a time average of the quantity in equation (5.71). With orbital energy and angular momentum no longer constant, this would be quite a complicated problem, and is not attempted.

Therefore, while predicting how dynamical friction will affect the velocity distribution of a GCS is a very complicated problem, inspection of equation (5.61) indicates that the effect will be more pronounced for more massive clusters in higher density environments, i.e. dynamical friction will most strongly affect massive clusters on low-energy orbits.

5.2.5 Evolution on the Globular Cluster Fundamental Plane

Recalling that GC properties are interrelated, and in fact can be uniquely represented with only two parameters, the GCFP can be represented in many ways (e.g., R_h vs L or σ vs L as in Figure 1.4, or M vs R_h as in Figure 1.11; cf. section 1.3). Consequently,

predicting how GCs evolve on the GCFP requires a theory built on the combined evolution of two such parameters. The destruction mechanisms listed above for evaporation, stellar evolution, tidal shocking and dynamical friction serve to predict how GC mass evolves with time. Therefore all that is required for a theory of GCFP evolution is a prescription detailing how another (non-equivalent) GC property changes with time in response to this mass-loss. Such a prescription may be derived based on considerations of internal energy change and the virial theorem as follows. In general the total internal energy per unit mass of a system is:

$$\tilde{E}_{tot} = \tilde{K}_{tot} + \tilde{W}_{tot} \quad (5.72)$$

where \tilde{K}_{tot} is the total kinetic energy per unit mass of all constituent particles, and \tilde{W}_{tot} is the total mutual gravitational binding energy per unit mass. If the system is virialised then the following must also be true (e.g. equation 1.1):

$$2\tilde{K}_{tot} + \tilde{W}_{tot} = 0 \quad (5.73)$$

Therefore, for the system to remain virialised when the total energy of the system is changed by some amount $\delta\tilde{E}$ in time $\delta\tilde{t}$, the total kinetic and potential energies must satisfy:

$$\delta\tilde{E}_{tot} = \delta\tilde{K}_{tot} + \delta\tilde{W}_{tot} \quad (5.74)$$

$$0 = 2\delta\tilde{K}_{tot} + \delta\tilde{W}_{tot} \quad (5.75)$$

Combining these, the following standard result is obtained:

$$\delta\tilde{K}_{tot} = -\delta\tilde{E}_{tot} \quad (5.76)$$

$$\delta\tilde{W}_{tot} = 2\delta\tilde{E}_{tot} \quad (5.77)$$

and therefore in the limit of infinitesimal changes ($\delta \rightarrow 0$), for the system to remain virialised these quantities must change at a rate:

$$\frac{d\tilde{K}_{tot}}{d\tilde{t}} = -\frac{d\tilde{E}_{tot}}{d\tilde{t}} \quad (5.78)$$

$$\frac{d\tilde{W}_{tot}}{d\tilde{t}} = 2\frac{d\tilde{E}_{tot}}{d\tilde{t}} \quad (5.79)$$

For a cluster undergoing mass-loss due to escapes, changes to the total energy will be brought about due to kinetic energy being carried away by escapers, changes to the gravitational potential due to mass-loss and spatial redistribution of stars, and binary reheating (see section 1.4.5). Ignoring binary reheating for simplicity, this implies that:

$$\frac{d\tilde{E}_{tot}}{d\tilde{t}} = \frac{d\tilde{K}_{esc}}{d\tilde{t}} + \frac{d\tilde{W}_{tot}}{d\tilde{t}} \quad (5.80)$$

where \tilde{K}_{esc} is the kinetic energy per unit mass carried away by escapers. Combining this with equations (5.78) and (5.79) implies that:

$$\frac{d\tilde{K}_{tot}}{d\tilde{t}} = \frac{d\tilde{K}_{esc}}{d\tilde{t}} \quad (5.81)$$

$$\frac{d\tilde{W}_{tot}}{d\tilde{t}} = -2\frac{d\tilde{K}_{esc}}{d\tilde{t}} \quad (5.82)$$

such that $2d\tilde{K}_{tot}/d\tilde{t} + d\tilde{W}_{tot}/d\tilde{t} = 0$ is satisfied at all times for any $d\tilde{K}_{esc}/d\tilde{t}$, in accordance with the virial condition. In general, the gravitational binding energy may be expressed in terms of the gravitational radius, \tilde{r}_g . However, for many systems the half-mass radius and gravitational radius are related via $\tilde{r}_h = C_W \tilde{r}_g$ with $C_w \simeq 0.45$ (see section 1.2.1), which can also be combined with the definition of half-mass density to arrive at:

$$\tilde{W}_{tot} = -C_W \left(\frac{2P_0}{3}\right)^{1/3} \left(\frac{\bar{m}_0}{M_0}\right)^{2/3} \tilde{\rho}_h^{1/3} \tilde{M}_c^{5/3} \tilde{m}^{-1} \quad (5.83)$$

and therefore:

$$\frac{d\tilde{W}_{tot}}{d\tilde{t}} = \tilde{W}_{tot} \left(\frac{5}{3\tilde{M}_c} \frac{d\tilde{M}_c}{d\tilde{t}} + \frac{1}{3\tilde{\rho}_h} \frac{d\tilde{\rho}_h}{d\tilde{t}} - \frac{1}{\tilde{m}} \frac{d\tilde{m}}{d\tilde{t}} \right) \quad (5.84)$$

Substituting in equation (5.82), the rate of change of half-mass density may be expressed as a function of mass-loss rate, rate of kinetic energy carried away by escapers, and rate of change of average stellar mass:

$$\frac{1}{\tilde{\rho}_h} \frac{d\tilde{\rho}_h}{d\tilde{t}} = \frac{3}{\tilde{m}} \frac{d\tilde{m}}{d\tilde{t}} - \frac{6}{\tilde{W}_{tot}} \frac{d\tilde{K}_{esc}}{d\tilde{t}} - \frac{5}{\tilde{M}_c} \frac{d\tilde{M}_c}{d\tilde{t}} \quad (5.85)$$

In order to proceed further, it is necessary to consider the kinematics of escapers in a cluster. The energy per unit mass of a particle in a rotating frame (known as the Jacobi integral) is given by:

$$\tilde{E}_J = \frac{1}{2} |\dot{\tilde{\mathbf{x}}}|^2 + \tilde{\Phi}(\tilde{\mathbf{x}}) - \frac{1}{2} |\tilde{\boldsymbol{\Omega}} \times \tilde{\mathbf{x}}|^2 \quad (5.86)$$

where $\tilde{\mathbf{x}}$ is the position vector, and $\tilde{\boldsymbol{\Omega}}$ is the angular frequency vector of the particle. Defining a coordinate system where the x -axis is aligned along the line joining the centre of mass of the cluster and host galaxy, the z -axis is perpendicular to the orbital plane, and the y -axis is perpendicular to both of these, then $\tilde{\boldsymbol{\Omega}} = (0, 0, \tilde{\Omega})$. Assuming that the mass interior to the tidal radius is sufficiently concentrated as to be well approximated by a point mass potential, the minimum energy per unit mass required to reach the tidal radius at \tilde{r}_t and escape from the cluster is:

$$\tilde{E}_{crit} = -\frac{3\tilde{M}_c}{2\tilde{r}_t} \frac{\tilde{m}_0}{M_0} = -\frac{3}{2} \left(\frac{P_0}{3} \right)^{1/3} \left(\frac{\tilde{m}_0}{M_0} \right)^{2/3} \tilde{\rho}_t^{1/3} \tilde{M}_c^{2/3} \quad (5.87)$$

and the kinetic energy carried away by an escaper is:

$$\tilde{E}_{pe} = \tilde{E} - \tilde{E}_{crit} \quad (5.88)$$

As discussed in section 5.2.1, escape is considered under two regimes; the energy criterion, where stars that achieve energies of \tilde{E}_{crit} are assumed to leave the cluster instantaneously, or the apocentre criterion, where once having achieved an energy of \tilde{E}_{crit} stars may have other encounters while in transit to \tilde{r}_t , and be scattered back below $\tilde{E} = \tilde{E}_{crit}$. Therefore, such stars may continue to have their energies up or down scattered before actually escaping, and consequently are called potential escapers. As

a result, a population of potential escapers is established over a range of energies, with a minimum of \tilde{E}_{crit} .

This process of up and down scattering in energies is diffusion through energy-space, with the number of potential escapers at energy \tilde{E}_{pe} determined by the difference in the rate that potential escapers are scattered to and from this energy, minus the rate that potential escapers at this energy actually escape. Therefore, the rate of change of the number of potential escapers per unit energy, $\eta_{pe} = dN_{pe}/d\tilde{E}_{pe}$, may be expressed according to Fick's second law including an additional term to take into account the rate that stars actually escape from the cluster:

$$\frac{d\eta_{pe}}{d\tilde{t}} = \alpha \frac{d^2\eta_{pe}}{d\tilde{E}^2} - \frac{\eta_{pe}}{\tilde{t}_{esc}} \quad (5.89)$$

The constant α is the diffusion coefficient, with dimensions of $\tilde{E}^2\tilde{t}^{-1}$. Since this process involves the number of particles scattered over \tilde{E}_{crit} per relaxation time, a natural choice for this constant is (e.g. Baumgardt 2001):

$$\alpha = \frac{\xi \tilde{E}_{crit}^2}{\tilde{t}_{rh}} = \frac{9\xi}{2^{3/2}} \left(\frac{P_0}{3}\right)^{7/6} \left(\frac{\bar{m}_0}{M_0}\right)^{4/3} \tilde{\rho}_t^{2/3} \tilde{\rho}_h^{1/2} \tilde{M}_c^{1/3} \tilde{m} \ln \left(\gamma \frac{\tilde{M}_c}{\tilde{m}} \right) \quad (5.90)$$

where ξ is the fraction of the velocity distribution that is scattered above the escape speed every relaxation time.

The more energy a potential escaper has, the more quickly it will be moving, and therefore the time taken for a potential escaper to leave the cluster will depend on how much excess energy above the escape energy it has. Fukushige & Heggie (2000) calculated this escape time to vary as:

$$t_{esc} = \frac{2C_\beta\sqrt{6}}{\pi} \frac{(GM_c)^{4/3}\omega^{1/3}}{(E - E_{crit})^2} \quad (5.91)$$

where $C_\beta \simeq 0.38$ and $\omega = \sqrt{\frac{GM_c}{3r_t^3}}$. This may be rewritten as:

$$\tilde{t}_{esc} = \beta \tilde{E}_{pe}^{-2} \quad (5.92)$$

$$\beta = \frac{2^{3/2}C_\beta}{\pi} \left(\frac{\bar{m}}{M_0}\right)^{4/3} (3P_0\tilde{\rho}_t)^{1/6} \tilde{M}_c^{4/3} \quad (5.93)$$

Note that β as calculated by Fukushige & Heggie (2000) provides an upper limit on the escape times, but that the energy scaling is robust (e.g. Takahashi & Baumgardt 2012). With the definition of \tilde{E}_{pe} given in equation (5.88), the differential equation may be written as:

$$\frac{d\eta_{pe}}{d\tilde{t}} = \alpha \frac{d^2\eta_{pe}}{d\tilde{E}_{pe}^2} - \tilde{E}_{pe}^2 \frac{\eta_{pe}}{\beta} \quad (5.94)$$

For a steady rate of mass-loss, it must be the case that $d\eta_{pe}/d\tilde{t} \simeq 0$, as $d\eta_{pe}/d\tilde{t} \ll 0$ implies that the rate at which potential escapers at energy \tilde{E}_{pe} actually escape far exceeds the rate at which potential escapers at this energy are populated by scatterings. Consequently, the population of potential escapers would be rapidly depleted, i.e. this would correspond to the energy criterion. Alternatively if $d\eta_{pe}/d\tilde{t} \gg 0$, then the rate at which potential escapers at energy \tilde{E}_{pe} are populated by scatterings far exceeds the rate at which potential escapers at this energy escape. As a result, the population of potential escapers would grow indefinitely while the number of bound stars would be depleted until the cluster became completely unbound and lost all member stars virtually instantaneously. Therefore although η_{pe} will likely change over the course of a cluster lifetime, these changes must occur on a time-scale that is very long compared to the time-scale for changes to individual potential escapers. Thus, $d\eta_{pe}/d\tilde{t} = 0$ should be a reasonable approximation (e.g. Baumgardt 2001):

$$\frac{d^2\eta_{pe}}{d\tilde{E}_{pe}^2} - A\tilde{E}_{pe}^2\eta_{pe} = 0 \quad (5.95)$$

with $A = (\alpha\beta)^{-1}$, given by (equations 5.90 and 5.93):

$$A = \frac{0.138\pi \left(\frac{M_0}{m_0}\right)^{8/3}}{3C_\beta \xi P_0^{4/3} \tilde{M}_c^{5/3} \tilde{\rho}_t^{5/6} \tilde{\rho}_h^{1/2} \tilde{m} \ln \left[\gamma \frac{\tilde{M}_c}{\tilde{m}} \right]} \quad (5.96)$$

The solution to equation (5.95) is given by (e.g. Baumgardt 2001):

$$\eta_{pe} = \frac{dN_{pe}}{d\tilde{E}_{pe}} = C_{pe} \left(\frac{A}{4}\right)^{1/8} \left(\frac{\tilde{E}_{pe}}{\pi}\right)^{1/2} K_{1/4} \left(\frac{A^{1/2}\tilde{E}_{pe}^2}{2}\right) \quad (5.97)$$

where K_n is the modified Bessel function of the second kind of order n , and C_{pe} is a normalisation constant. This constant can be calculated by requiring that equation (5.97) integrated over all \tilde{E}_{pe} gives the total number of potential escapers, N_{pe} , resulting in:

$$C_{pe} = \frac{N_{pe}}{\Gamma(5/4)} \left(\frac{A}{2} \right)^{1/4} \quad (5.98)$$

$$\eta_{pe} = \frac{dN_{pe}}{d\tilde{E}_{pe}} = \frac{A^{3/8} N_{pe}}{\Gamma(5/4)} \left(\frac{\tilde{E}_{pe}}{2\pi} \right)^{1/2} K_{1/4} \left(\frac{A^{1/2} \tilde{E}_{pe}^2}{2} \right) \quad (5.99)$$

where $\Gamma(n)$ is the Gamma function. The rate at which potential escapers at energy \tilde{E}_{pe} escape is given by $-\eta_{pe}/\tilde{t}_{esc}$:

$$\frac{d^2 N_{esc}}{d\tilde{t} d\tilde{E}_{pe}} = -\frac{A^{3/8} N_{pe}}{\Gamma(5/4)\beta} \frac{\tilde{E}_{pe}^{5/2}}{(2\pi)^{1/2}} K_{1/4} \left(\frac{A^{1/2} \tilde{E}_{pe}^2}{2} \right) \quad (5.100)$$

The rate at which escapers at this energy carry away energy is then simply the energy of each escaper, \tilde{E}_{pe} , multiplied by this rate of escape:

$$\frac{d^2 \tilde{K}_{esc}}{d\tilde{t} d\tilde{E}_{pe}} = -\frac{A^{3/8} N_{pe}}{\Gamma(5/4)\beta} \frac{\tilde{E}_{pe}^{7/2}}{(2\pi)^{1/2}} K_{1/4} \left(\frac{A^{1/2} \tilde{E}_{pe}^2}{2} \right) \quad (5.101)$$

And consequently, the total rate of energy lost due to escapers is obtained by integrating equation (5.101) over all \tilde{E}_{pe} , leading to:

$$\frac{d\tilde{K}_{esc}}{d\tilde{t}} = -\frac{N_{pe}}{2\pi^{1/2}\beta A^{3/4}} \frac{\Gamma(1/4)}{\Gamma(5/4)} \quad (5.102)$$

At this point all that is required is N_{pe} . This may be obtained by requiring that the rate of escapes at energy \tilde{E}_{pe} (equation 5.100) integrated over all \tilde{E}_{pe} gives the total rate of escapes, leading to:

$$\begin{aligned} N_{pe} &= -\frac{\Gamma(5/4)}{\Gamma(3/4)} (A\pi)^{1/2} \beta \frac{dN_{esc}}{d\tilde{t}} \\ &= -\frac{\Gamma(5/4)}{\Gamma(3/4)} (A\pi)^{1/2} \beta \frac{\tilde{M}_c}{\tilde{m}} \left(\frac{1}{\tilde{M}_c} \frac{d\tilde{M}_c}{d\tilde{t}} - \frac{1}{\tilde{m}} \frac{d\tilde{m}}{d\tilde{t}} \right) \end{aligned} \quad (5.103)$$

and therefore the rate of loss of kinetic energy per unit mass due to escapers is given by:

$$\frac{d\tilde{K}_{esc}}{d\tilde{t}} = \frac{\Gamma(1/4)}{2A^{1/4}\Gamma(3/4)} \frac{\widetilde{M}_c}{\tilde{m}} \left(\frac{1}{\widetilde{M}_c} \frac{d\widetilde{M}_c}{d\tilde{t}} - \frac{1}{\tilde{m}} \frac{d\tilde{m}}{d\tilde{t}} \right) \quad (5.104)$$

Thus, substituting in the definitions of \widetilde{W}_{tot} (equation 5.83) and A (equation 5.96), the rate of change of half-mass density may finally be written as:

$$\frac{d\tilde{\rho}_h}{d\tilde{t}} = \frac{\tilde{\rho}_h}{\tilde{m}} \frac{d\tilde{m}}{d\tilde{t}} [3 - \kappa] + \frac{\tilde{\rho}_h}{\widetilde{M}_c} \frac{d\widetilde{M}_c}{d\tilde{t}} [\kappa - 5] \quad (5.105)$$

$$\kappa = \left(\frac{81}{2} \right)^{1/3} \frac{\Gamma(1/4)}{C_W \Gamma(3/4)} \left(\frac{3C_\beta \xi \ln \left[\gamma \frac{\widetilde{M}_c}{\tilde{m}} \right]}{0.138\pi \frac{\widetilde{M}_c}{\tilde{m}}} \right)^{1/4} \left(\frac{\tilde{\rho}_t}{\tilde{\rho}_h} \right)^{5/24} \quad (5.106)$$

where $d\tilde{m}/d\tilde{t}$ is given by equation (5.56) when stellar evolution is desired, $d\widetilde{M}_c/d\tilde{t}$ is given by the choice of destruction mechanisms, $C_W \simeq 0.45$, $C_\beta \simeq 0.38$, $\gamma = 0.02$, and since the dependence on ξ is very weak, it is fixed at $\xi = 0.078$ (see Chapter 2). The method with which $\xi = 0.078$ was obtained did not explicitly treat evaporation mass-loss under the apocentre criterion, whereas the derivation of equation (5.106) does. However, ξ was obtained as a fitting parameter through χ^2 minimisation, and therefore the value of ξ is that which most closely matches the assumed evaporation mass-loss rate to the mass-loss rate (by any mechanism) that Virgo GCs actually underwent in turning power-law CIMFs into the GCMFs observed today. Given the fact that the models produced with the assumed evaporation mass-loss fit the observed GCMFs closely, ξ is likely to be a reasonable estimation of the actual mass-loss rate (see Chapter 2 for a more thorough discussion).

Ignoring any dependence on stellar evolution for simplicity, since $d\widetilde{M}_c/d\tilde{t} < 0$, equation (5.105) predicts that the half-mass density of a cluster increases when $\kappa < 5$, and decreases when $\kappa > 5$. If the half-mass density decreases, then κ will increase (cf. equation 5.106), which will result in further decrease of the half-mass density. On the

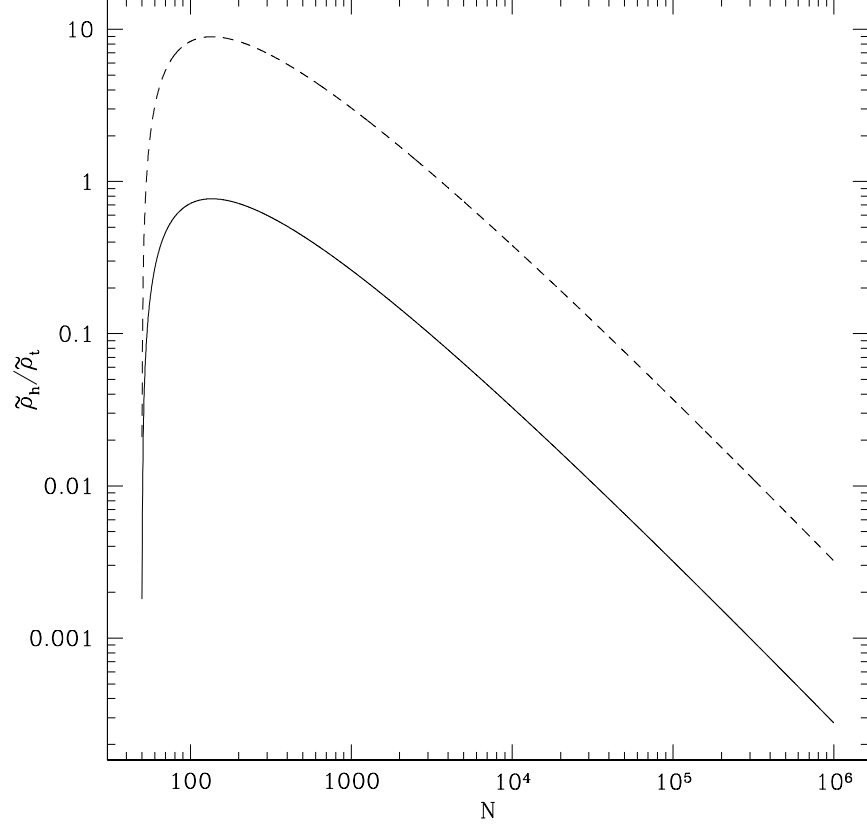


Figure 5.3: Ratio of half-mass to tidal density vs number of constituent particles, $N = \widetilde{M}_c/\tilde{m}$, for constant κ (cf. equation 5.106). The solid line corresponds to $\kappa = 5$, while the dashed line corresponds to $\kappa = 3$. The constants have values $C_W = 0.45$, $C_\beta = 0.38$, $\xi = 0.078$, and $\gamma = 0.02$.

other hand, the effect of decreasing mass is to increase κ , and therefore any increase in half-mass density must overcome this effect to decrease κ . This will likely be easier for clusters that already have high half-mass densities, as equation (5.105) predicts larger $d\tilde{\rho}_h/d\tilde{t}$ for larger $\tilde{\rho}_h$ at fixed $d\widetilde{M}_c/d\tilde{t}$, i.e. the same loss of mass will result in a larger gain in density during an interval of time $d\tilde{t}$. Therefore, qualitatively $d\tilde{\rho}_h/d\tilde{t} = 0$ is unstable, and high $\tilde{\rho}_h$ clusters will continue to evolve towards higher densities, while clusters with low $\tilde{\rho}_h$ will continue to evolve towards ever lower densities. Figure 5.3

displays lines of constant $\kappa = 3, 5$ (dashed, solid) in $\tilde{\rho}_h/\tilde{\rho}_t$ vs $N = \tilde{M}_c/\tilde{m}$ (note that the steep descent at small N is due to the Coulomb logarithm, which tends to zero as N approaches $1/\gamma = 50$). Clusters lying above the dashed line would have $\kappa < 3$, and consequently stellar evolution would act to reduce cluster density, while mass-loss would act in opposition to increase cluster density. Clusters lying between the dashed and solid lines would have $3 < \kappa < 5$, and thus both stellar evolution and mass-loss would act in concert to increase cluster density. Alternatively, clusters that lie below the solid line would have $\kappa > 5$, and therefore stellar evolution would act to increase cluster density, with mass-loss acting in opposition. Continuing to ignore stellar evolution for simplicity, any clusters that lie above the solid line will be evolving towards higher densities, whilst those that lie below the solid line will evolve towards lower densities. Since the peak in this curve occurs at $\tilde{\rho}_h/\tilde{\rho}_t \simeq 0.8 \Rightarrow \tilde{r}_h/\tilde{r}_t \simeq 0.85$, the vast majority of clusters will lie above the solid line. For example, King model clusters are roughly bound between $\tilde{r}_h/\tilde{r}_t = [0.1, 0.35]$ (cf. Figure 1.7), which approximately corresponds to $\tilde{\rho}_h/\tilde{\rho}_t = [12, 500]$. In fact, the meaning of $\tilde{\rho}_h/\tilde{\rho}_t < 1$ is ambiguous, since in terms of only bound stars this would imply a cluster density profile that increases outwards. Alternatively this could be interpreted to mean that a cluster is overfilling its Roche lobe, and consequently must be undergoing severe mass-loss. Consequently the expectation is that a GCS evolved according to this prescription will produce a GCS with a large spread in densities, as some GCs continuously evolve towards lower densities, and others continuously evolve towards higher densities. The fact that some clusters will tend towards ever higher densities is very reminiscent of core collapse, which is actually eventually halted by binary reheating in the core (see section 1.4.5). A more detailed derivation of equation (5.105) including a term for binary reheating in step (5.80) would very likely result in a prescription for density evolution predicting a similar eventual halt to the increase in density, and possibly even gravothermal oscillations.

In this Chapter, a large number of evaporation mass-loss prescriptions have been collected, and additionally prescriptions for tidal shocking mass-loss and dynamical

friction have also been taken from the literature and discussed. Furthermore, prescriptions for stellar evolution mass-loss and cluster half-mass density evolution have been derived. Each of these categories of prescription were discussed in terms of what affect they are likely to have on an evolved GCS. Furthermore, the multiple prescriptions in the evaporation mass-loss and stellar evolution mass-loss were compared, and the likely differences in the GCS arising from evolution with these different prescriptions discussed. Examples of simulated GCSs evolved from the initial GCSs of Chapters 3 and 4 using the prescriptions presented in this Chapter are displayed next in Chapter 6.

6 Examples of Simulated Evolved GCSs

In this Chapter, the techniques, methods, and prescriptions presented and discussed in Chapter 5 are applied to simulated initial GCSs created according to the method presented in Chapter 3 and displayed in Chapter 4, to create simulated evolved GCSs. These evolved GCSs are presented (where relevant) in terms of the properties discussed in section 1.3, namely the mass function, velocity distribution, and fundamental plane. Note that all of these models are merely demonstrations that the evolution routine and prescriptions work as expected, and are not intended to be representative of real GCSs. Before these plots are displayed and discussed however, the run-time commands of the GCS evolution code are described, in order to demonstrate the versatility of the methods discussed in Chapter 5.

6.1 Simulated GCS Evolution Code Description

In order to allow easy comparisons between GCSs evolved from the same initial set-up with different evolutionary prescriptions, the set-up of the initial GCS and the GCS evolution are performed separately by different codes. The operation of the first code is described in section 3.7. The second code applies the equations and methods given in this Chapter to evolve the simulated initial GCS from the first code to provide an evolved simulated GCS (see Appendix F). The run-time commands are described below.

First, the desired final GCS age is required, followed by the ratio of the initial stellar mass, m_0 , to the host galaxy mass-scale, M_0 . This roughly controls how much more massive the host galaxy is than a typical tracer. Next to be selected are the prescriptions; evaporation (see section 5.2.1), tidal shocking (see section 5.2.2), stellar evolution (see section 5.2.3), dynamical friction (see section 5.2.4), and half-mass density evolution (see section 5.2.5), with the possibility of any being switched off. Of these five mechanisms, only stellar evolution requires any additional input. This is

because the maximum and minimum stellar IMF mass, along with the probability of retention of a newly formed remnant for each remnant class (i.e. white dwarfs, neutron stars, and black holes) are free parameters to be input. Since mass-loss due to stellar evolution is expected to dominate early on in cluster evolution, and stellar evolution affects all clusters equally, modelling this stage of evolution would basically only serve to reduce the initial mass of all clusters before more interesting effects such as evaporation and tidal shocking become important (see section 1.4.1). For this reason it is assumed that this phase of extreme mass-loss due to stellar evolution has already passed for the initial GCS produced as described in Chapter 3, i.e. the initial GCS begins with an age of $\tilde{t} = \tilde{t}_{start} = 30$. Thus, when selecting the limits of the stellar IMF, m_l and m_u , the upper mass limit must satisfy $m_u \leq m_{to}(\tilde{t}_{start})$.

Next, the user is prompted for a definition of the effective tidal radius, \tilde{r}_{ev} , i.e., at which galactocentric radius the corresponding tracer tidal radius matches the average tracer tidal radius determining dynamical evolution (see section 1.4.4). The options are:

1. Orbital pericentre
2. Time averaged galactocentric radius
3. Galactocentric radius at which the square root of instantaneous tidal density equals the time average of the square root of tidal density
4. Galactocentric radius where the tidal density equals a specific function of ellipticity multiplying the tidal density at orbital pericentre

The standard approach is to define the effective tidal radius at orbital pericentre, however as discussed in section 1.4.4, this may be an oversimplification. The reasoning behind setting effective tides at orbital pericentre is because a typical orbital time-scale is much shorter than a typical cluster relaxation time, and therefore as the outer layers of a cluster are stripped by tides at perigalacton, there is not enough time for substantial expansion before the next perigalactic passage (von Hoerner 1957). However, given that

technically unbound stars may continue to share similar orbits to their former host for a very long time (Fukushige & Heggie 2000), it is possible that as a cluster moves away from the galactic centre and the instantaneous tides weaken, the growing tidal radius could envelop some of the recently ‘shed’ stars, and they may become bound once again (e.g. Webb et al. 2013). Consequently, the mass-loss of the cluster is not as severe as would be expected at pericentre, and the orbital radius corresponding to the effective tides must lie farther out than pericentre (see also section 1.4.4). Webb et al. (2013) sought to account for this, and by fitting to N-body simulations derived a correction to the pericentric tides giving the tidal density corresponding to the mass-loss rate at the current orbital radius. This is the function of ellipticity mentioned in the final option for definition of effective tides:

$$\tilde{\rho}_t(\tilde{r}_M) = \tilde{\rho}_t(\tilde{r}_p) [1 + aF \exp(be)]^{-3} \quad (6.1)$$

where \tilde{r}_p is the orbital pericentre, $a \simeq 0.17$, $b \simeq 4.1$, $F = \frac{\tilde{r}_M - \tilde{r}_p}{\tilde{r}_a - \tilde{r}_p}$ is the orbital phase, and e is the orbital ellipticity.

Following the definition of effective tidal radius, the user is prompted to choose whether the half-mass densities of clusters will be allowed to evolve or not. If not, then the integration variable for half-mass density is fixed, and the evolution integrations are a system of four equations, three for orbital motion (two of which are given by the radial motion due to it being a second derivative, cf. equation (5.9), while the third is given by equation (5.10), describing tangential motion). The fourth evolution equation is the time derivative of cluster mass, $d\tilde{M}_c/d\tilde{t}$, which is the sum of mass-loss rates due to any enabled prescriptions listed in section 5.2. When the option for the half-mass density of clusters to evolve in response to mass-loss according to maintained virial equilibrium is selected (see section 5.2.5), the fifth evolution equation is given by equation (5.105). Subsequently, a choice of how to assign initial half-mass densities to the tracer population is presented. Although most evaporation prescriptions do not depend on half-mass densities, they are essential for tidal shocks and evolution of the half-mass density, and also make a difference to dynamical friction through the Coulomb logarithm (see section 5.2.4). The options are:

1. To treat tracers as if they were point masses, however this is not allowed if a prescription requiring a non-zero half-mass density has been selected (e.g. tidal shocking, half-mass density evolution, or certain evaporation prescriptions).
2. To randomly draw a King model concentration from either a Gaussian or Schechter (1976) distribution. If a Gaussian is selected to describe the concentration distribution, the mean and variance must also be specified, whereas if a Schechter (1976) function is selected to describe the concentration distribution, the exponential cut-off and logarithmic power-law slope must also be specified. Whichever concentration distribution is selected, a minimum and maximum allowed concentration is also required, to prevent extreme outliers, in addition to a seed for the random number generator. Once a concentration has been assigned, it is used to obtain a value of $\tilde{\rho}_h/\tilde{\rho}_t$. Then assuming that \tilde{r}_t is the tidal radius at the time averaged galactocentric radius, $\tilde{\rho}_h$ is solved for.
3. To specify a relation between the ratio of half-mass radius to tidal radius and cluster mass, i.e. $\tilde{r}_h/\tilde{r}_t = A\tilde{M}_c^B$, where A and B are input by the user, and \tilde{r}_t is calculated using the previously selected definition of effective tides to obtain $\tilde{\rho}_h$.
4. To specify a relation between the half-mass radius and cluster mass directly; $\tilde{r}_h = A\tilde{M}_c^B$, with A and B input by the user.

When an evaporation prescription assuming a King (1966) cluster structure is selected (i.e. with an explicit dependence on W_0 , namely the King (1966) and Lamers, Baumgardt & Gieles (2010) evaporation prescriptions), only the second option assigning a random concentration to all clusters is allowed. This is because \tilde{r}_h/\tilde{r}_t is multivalued in c and thus one value of \tilde{r}_h/\tilde{r}_t may correspond to several values of c and W_0 (since W_0 and c have a one-to-one relation; King 1966). Therefore, a random value of c is assigned first, and interpolation is used to find W_0 and \tilde{r}_h/\tilde{r}_t . Otherwise, if any other selected prescriptions depend on half-mass density (such as tidal shocking, various evaporation

prescriptions, half-mass density evolution, etc.), then any choice other than treating tracers as point masses is allowed.

The options for dynamical friction are either Chandrasekhar (1942) or none at all. If Chandrasekhar (1942) was selected, then the orbital energy and angular momentum of each tracer is no longer conserved, meaning that all tracers will migrate to different orbits at each instant in time as their orbital energies and angular momenta decay. Consequently, all associated orbital parameters (i.e. orbital pericentre, apocentre, etc.) are no longer constant with time. Thus the orbital parameters of a tracer at a single instant in time are the orbital parameters corresponding to the tracer's energy and angular momentum at that instant in time. Therefore, for every energy between $\tilde{E} = \tilde{\Phi}(\tilde{r}_{init})$ and $\tilde{E} = \tilde{\Phi}(\tilde{r}_f)$, all required orbital parameters are calculated as functions of the parameter $l = [\tilde{L} - \tilde{L}_{min}(\tilde{E})]/[\tilde{L}_c(\tilde{E}) - \tilde{L}_{min}(\tilde{E})]$ between $l = 0 \Rightarrow \tilde{L} = \tilde{L}_{min}(\tilde{E})$ and $l = 1 \Rightarrow \tilde{L} = \tilde{L}_c(\tilde{E})$. This is slightly different from the definition of h in equation (3.2) in that l depends on $\tilde{L}_{min}(\tilde{E})$ in place of \tilde{L}_0 , where $\tilde{L}_{min}(\tilde{E})$ is the angular momentum required to have a pericentre at $\tilde{r}_{min} = 10^{-4}$, i.e.

$$\tilde{L}_{min} = \sqrt{2 \left(\tilde{E} - \tilde{\Phi}(\tilde{r}_{min}) \right) \tilde{r}_{min}} \quad (6.2)$$

The purpose of this definition of l is twofold; firstly, the calculation of parameters such as pericentre, apocentre, radial period, and orbital averages of certain quantities become intractable for very radial orbits, and in reality a GC with a pericentre of $r_p = 10^{-4}r_0$ would be deep inside the core of the host galaxy, and would be very unlikely to ever re-emerge. Consequently, disallowing such orbits does not detract in any significant way from the potential utility of the evolution code. The second reason for the definition of l is a numerical one. The orbital circularity is always bounded by $0 \leq h \leq \tilde{L}/[\tilde{L}(\tilde{E}) + \tilde{L}_0]$, and so the upper limit varies with orbital energy when $\tilde{L}_0 > 0$. This would make interpolating on tables of orbital parameters as functions of \tilde{E} and h unnecessarily difficult. On the other hand, l is always bounded by $0 \leq l \leq 1$ for any \tilde{E} , and therefore all required orbital parameters can be easily interpolated for, given instantaneous values of \tilde{E} and \tilde{L} . In practice, it is also very difficult to solve

for the orbital parameters of nearly circular orbits. For example, as $\tilde{L} \rightarrow \tilde{L}_c(\tilde{E})$, the difference between pericentre and apocentre becomes smaller than numerical noise, and it is possible for the situation to arise where bisection will claim to have obtained an orbital pericentre greater than the orbital apocentre. Consequently any further calculations that depend on these values (such as equation 3.75) will go awry. Thus, the range over l for which orbital parameters are solved is slightly more conservative, and $\epsilon \leq l \leq 1 - \epsilon$ is adopted, with $\epsilon = 10^{-4}$. Therefore before the main orbital integrations, orbital pericentres, apocentres, time averaged radii, and radial periods are solved for all \tilde{E} and l on a grid of 60×240 to enable interpolation for these parameters, rather than calculating them on-the-fly during the main orbit integrations. On the very rare occasions when a tracer is on an orbit corresponding to l outside of the range $\epsilon \leq l \leq 1 - \epsilon$, logarithmic extrapolation is utilised. For an orbit with $l = 1 - \epsilon$, this corresponds to $\tilde{L}/\tilde{L}_c(\tilde{E}) = 1 - \epsilon[1 - \tilde{L}_{min}(\tilde{E})/\tilde{L}_c(\tilde{E})]$, and therefore no orbit with $l > 1 - \epsilon$ will ever be greater than 0.1% more than the largest \tilde{L} orbital parameters are tabulated for. Consequently logarithmic extrapolation deals with these rare situations reliably.

Once all the required information has been entered, the equations of motion of every tracer are integrated from their initial values at $\tilde{t} = \tilde{t}_{start} = 30$ up to the specified evolved tracer system age by use of a fourth order Runge-Kutta variable step size integration scheme (e.g. Press et al. 1992). Typically, (when dynamical friction is not active), orbital angular momenta are conserved extremely well over the course of the integration (any loss of precision is indistinguishable from numerical noise), while orbital energies are typically conserved to within a few percent (though this can be improved to a few tenths of a percent by requiring a higher accuracy from the integrations, with the trade-off of longer run-times). When dynamical friction is enabled, the equations of motion become much more difficult to solve, taking significantly longer. Despite the fact that dynamical friction is supposed to reduce orbital energy and angular momentum, the complicated nature of the integrations occasionally cause the integrator to step the integration variables in an unphysical way, which may cause unwanted behaviour such as an orbital angular momentum greater than $\tilde{L}_c(\tilde{E})$, or orbital

energy greater than the escape energy, \tilde{E}_{max} , etc. When this happens, the integration is aborted and the tracer in question is destroyed. Orbits which are already near to these limits are most vulnerable to this happening, and consequently having dynamical friction enabled will bias a GCS towards smaller orbits more strongly than the physical process is directly responsible for.

During the integration of the equations of motion, the selected prescriptions are applied to the tracer mass, density, orbital energy, and orbital angular momentum. Thus, the final values for tracer mass, galactocentric distance, orbital energy and angular momentum, and radial azimuthal and polar components of velocity are obtained for each tracer. The evolved GCS may then be compared directly to the initial GCS in terms of mass function, velocity distribution, and fundamental plane to reveal the exact details how the selected prescriptions have affected the GCS. Examples are given below for the performance of the evolution code with at least one of every prescription, and with various different options for CIMF, initial half-mass densities, etc. All GCSs are evolved to an age of 13 Gyr.

6.2 Evolved Evaporation Models

The first set of models displays GCSs evolved with only evaporation prescriptions, namely equations (5.16) and (5.20). Figure 6.1 displays the initial and final mass functions (black and red points) of a GCS in a $k = 2.5$ polytrope host galaxy. The CIMF of this GCS was a Schechter function (equation 3.63) with $\tilde{M}_{sch} = 10^6$ and $\beta_{MF} = 2$, and the initial half-mass densities of the GCs were assigned according to the relation $\tilde{r}_h = A\tilde{M}_c^B \Rightarrow \tilde{\rho}_h = \frac{\tilde{m}_0}{M_0} \frac{3\tilde{M}_c^{1-3B}}{2P_0A^3}$, with $A = 2.75 \times 10^{-4}$ and $B = 0.615$. These values for A and B were proposed by Gieles, Baumgardt & Heggie (2010), who argued for the existence of a global mass-radius relation common to both galaxies and clusters (an alternate form of the Faber-Jackson relation). This is essentially just the fit to galaxy data in Figure 1.4 extrapolated down to GC masses. Gieles, Baumgardt & Heggie (2010) argued that the vastly smaller masses of clusters means that they are

much more dynamically evolved, and consequently clusters have evolved away from this initial mass-radius relation through contraction/expansion and mass-loss. Since this initial mass-radius relation imposes a higher density on low-mass clusters, and the evaporation mass-loss rate of equation (5.16) is stronger for more dense clusters, the low-mass end of the GCMF has been strongly depleted. Meanwhile, the high-mass end of the GCMF remains relatively untouched, causing a distinct peak to emerge at around $\widetilde{M}_c \simeq 10^4$ after 13 Gyr of evaporation. The dimensionless tracer mass is related to real mass through $\widetilde{M}_c = M_c/\bar{m}_0$, where \bar{m}_0 is the initial average stellar mass in the cluster (a quantity that purely depends on the stellar IMF, and the maximum and minimum masses, m_u and m_l). For any realistic stellar IMF, \bar{m}_0 will not be far from $1 M_\odot$ for sensible choices of m_l and m_u (a Chabrier (2003) disk IMF with $m_l = 0.501 M_\odot$ and $m_u = 6.519 M_\odot$ has an initial average stellar mass of $\bar{m}_0 = 1.267 M_\odot$). Thus, the evolved GCS in units of M_\odot is about a factor of 1.25, or 0.01 dex higher than what is displayed.

While the peak of Figure 6.1 is slightly low compared to the peak masses of GCMFs such as the Milky Way and Virgo galaxies (see Chapter 2), again these models are intended solely as demonstrations, not to reflect real GCSs. However, decreasing A would result in higher initial densities by a constant factor for all masses, which would cause greater subsequent mass-loss due to evaporation at all masses. Alternatively, decreasing B would also increase initial densities, but by a greater amount for higher masses. Thus, it is not difficult to imagine that some slight adjustments to A and B would bring the peak mass more in line with the expected $M_{TO} \simeq 10^5 M_\odot$. However, this would also imply clusters with densities far higher than what is observed in the Milky Way and other galaxies (e.g. McLaughlin & van der Marel 2005). Therefore an alternative would be to use a higher value for the value of \tilde{C} in equation (5.16), such as that of McLaughlin & Fall (2008) for example. This is discussed further in Chapter 7.

By allocating initial half-mass densities based only on cluster initial masses and utilising an evaporation prescription that depends only on half-mass density, there was no primordial radial dependence built into the GCS, and no radial dependence

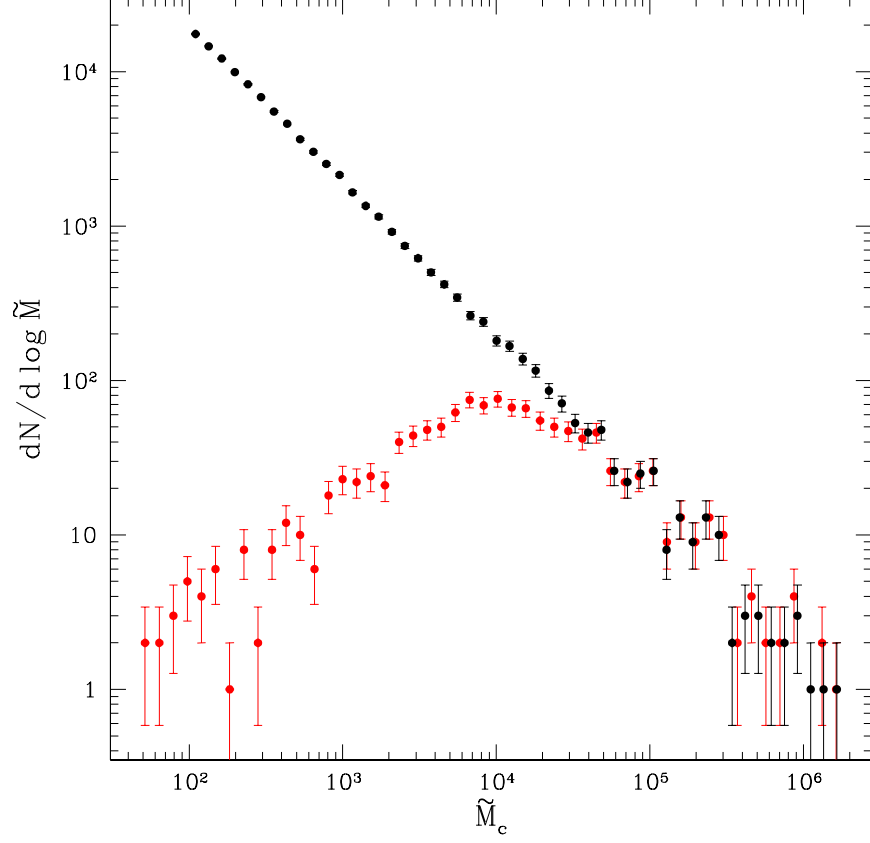


Figure 6.1: The initial (black points) and final (red points) mass functions of an initially isotropic GCS in a $k = 2.5$ polytrope host galaxy. The mass function was initially a Schechter CIMF, and the initial half-mass radius and initial mass were related according to $\tilde{r}_h = A\tilde{M}_c^B$. The only active evolution prescription was evaporation, given by equation (5.16).

was introduced through subsequent evolution and GC destruction. This lack of any radial dependence also means that the GCMF is completely radially invariant, and furthermore, this evolved GCS would be identical regardless of the host galaxy it was evolved in. However, one weakness of this model is that the evolved GCFP still reflects the initial mass-radius relation. This is because the half-mass densities were assigned according to initial mass, and assumed to be constant thereafter. Therefore since the

mass-loss rate utilised depends only on half-mass density, the mass-loss experienced by a cluster depends on its initial mass. Consequently, there is a very clear one-to-one relation between half-mass density and final mass, a feature which is completely absent in real GCSs (see section 1.3.1).

Figure 6.2 displays the GCMF of a GCS in a $k = 2.5$ polytrope host galaxy that began with a Gaussian CIMF (equation 3.64) with mean at $\widetilde{M}_c = 10^5$ and variance $\sigma_{\widetilde{M}_c} = 1$, and was subsequently evolved for 13 Gyr with evaporation prescription given by equation (5.20). The initial half-mass densities of GCs were assigned by drawing a random King model concentration from a Gaussian distribution with mean $\bar{c} = 2.15$ and variance $\sigma_c = 1$. The randomly assigned c then corresponds to a ratio $\tilde{\rho}_h/\tilde{\rho}_t$, which is then combined with the tidal density at the time-averaged orbital radius of the GC in question to give $\tilde{\rho}_h$. Consequently, primordial radial dependence was built into the GCS, as a larger orbit would have a larger time averaged radius, which would lead to a smaller tidal density, and thus a smaller half-mass density for the same c . Obviously this radial dependence is not apparent in the initial GCMF in Figure 6.2, as the initial radial dependence is in terms of half-mass density, not GC mass. The evaporation prescription selected was that of Gieles & Baumgardt (2008) under the apocentre criterion, such that the evaporation mass-loss rate depends on \widetilde{M}_c (both directly and through the Coulomb logarithm), $\tilde{\rho}_t$, and $\tilde{\rho}_h$ through dependence on cluster structure. The effective tides were set to operate from orbital pericentre, i.e. $\tilde{r}_{ev} = \tilde{r}_p$, and thus mass-loss due to evaporation also imparted radial dependence on the GCMF.

The different panels of Figure 6.2 display the evolved GCMF in different ranges of the orbital pericentres of the constituent GCs. The radial dependence of the GCMF is weak but discernible, with the peak of the GCMF at higher masses for smaller \tilde{r}_p . This is as expected since in this case, smaller \tilde{r}_p correspond to higher $\tilde{\rho}_h$ (and $\tilde{\rho}_t$, although in this evaporation prescription, $\tilde{\rho}_t$ only appears as a ratio with $\tilde{\rho}_h$, and since the sampling of c was radially independent, $\tilde{\rho}_t/\tilde{\rho}_h$ is also independent of radius). Consequently, clusters at smaller radii have lost more mass, and the peak has shifted to higher masses. Note that the rising number of evolved GCs at very low masses in all three panels is just where initially higher mass clusters have migrated to lower

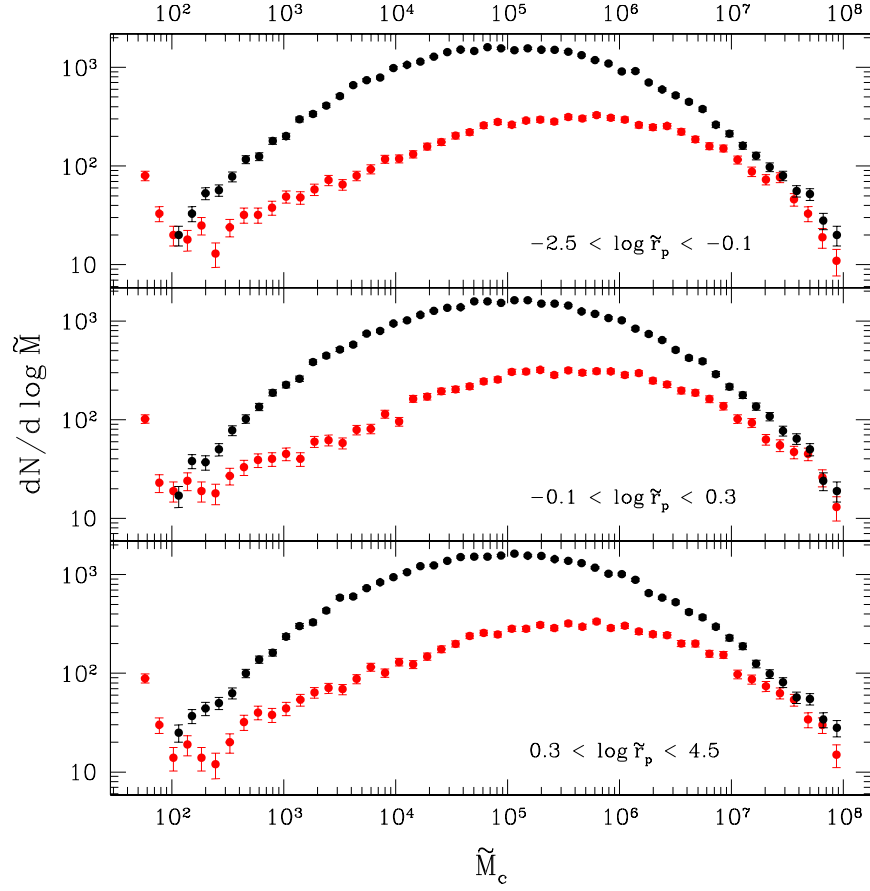


Figure 6.2: The initial (black points) and final (red points) mass functions of a GCS in a $k = 2.5$ polytrope host galaxy. The mass function was initially a Gaussian CIMF, and the initial half-mass densities of GCs were assigned by randomly drawing from a Gaussian of King model concentrations. The only active evolution prescription was evaporation, given by equation (5.20).

mass bins due to evaporation mass-loss, and have ended up in bins below the minimum mass of the CIMF. The criterion by which clusters are judged to be destroyed is when $\gamma N \leq 1$, where $\gamma = 0.02$ is the coefficient in the Coulomb logarithm (Giersz & Heggie 1994; Giersz & Heggie 1996), and N is the number of cluster stars. This was because for smaller N , the Coulomb logarithm will turn negative and cause evaporation mass-loss to add mass! Aside from being unphysical, this would be incorrect as the standard

form for the Coulomb logarithm of $\ln(\gamma N)$ is an approximation made for large N (e.g. Binney & Tremaine 2008). Therefore rather than revert to the full form of the Coulomb logarithm, $N = 50$ was assumed to be a reasonable cut-off. Since in this case stellar evolution was not active, $\widetilde{M}_c = N$, and thus the minimum mass was $\log \widetilde{M}_c \simeq 1.7$, whereas the minimum mass of the CIMF was $\log \widetilde{M}_c = 2$. Realistically, it would be natural to expect a newly formed GCS to be populated down to the minimum mass possible, rather than with an arbitrary cut-off imposed, and thus ideally the CIMF should also decrease down to $\widetilde{M}_c = 50$. Therefore, as clusters lose mass, they migrate to lower mass bins which were originally empty according to the CIMF. When coupled with the fact that the evaporation prescription employed in this case was derived under the apocentre criterion, the evaporation mass-loss rate is weaker for low-mass GCs, and furthermore the Coulomb logarithm cannot be neglected in this case, since as $\widetilde{M}_c \rightarrow \gamma^{-1}$, $\ln(\gamma \widetilde{M}_c) \rightarrow 0$. As a result, there is a build-up of very low-mass clusters with very low rates of mass-loss due to evaporation. Although less noticeable, the evolved GCMF in Figure 6.1 also has bins containing low-mass clusters that have migrated below the lowest mass bin of the CIMF. In this case however, the evaporation mass-loss rate depends only on cluster half-mass density and is therefore constant. Thus the evaporation mass-loss rate does not decrease as cluster mass decreases, and low-mass clusters continue to be destroyed efficiently, rather than accumulating as in Figure 6.2. Accumulations of low-mass GCs like this are not observed in extant GCSs, however such low masses would be very faint and thus hard to detect. In any event, once again these models are intended solely as demonstrations, rather than to resemble reality.

6.3 Evolved Tidal Shock Models

Figure 6.3 displays properties of an initially isotropic GCS with a Schechter CIMF that directly traced a $\gamma = 1$ Dehnen model host galaxy. Initial half-mass densities were assigned by drawing a random King model concentration from a Gaussian with mean $\bar{c} = 2.15$ and variance $\sigma_c = 1$. Each randomly assigned c was then combined with

the tidal density at each tracer’s time averaged orbital radius to obtain a half-mass density. The only evolutionary mechanism enabled was the tidal shocking prescription given by equation (5.50). In the top panel are displayed the initial (black) and final (red) mass functions, in the middle panel are the initial and final velocity anisotropy curves, while in the bottom panel is the mean half-mass density of tracers binned by orbital pericentre plotted against orbital pericentre.

Tidal shocks are only expected to significantly affect clusters with short radial periods, high masses, low densities, or small orbital pericentres (see section 5.2.2). By comparing the initial (black) and final (red) mass functions displayed in the top panel of Figure 6.3 some minor mass-loss is noticeable. The reason that tidal shocking has been so ineffective in this model is as follows; clusters on small orbits will have high tidal densities, which when combined with the randomly sampled King model concentration will correspond to high half-mass densities, as displayed in the bottom panel. Thus, since this GCS is quite centrally concentrated, statistically a randomly sampled cluster is likely to have a high half-mass density, and therefore tidal shocking will be inefficient. When combined with the increasing statistical noise towards higher masses, determining whether tidal shocking has more significantly affected high-mass clusters is practically impossible. Furthermore, the mass-loss is not sufficiently severe for any obvious trends to emerge when the GCMF is divided into groups based on orbital pericentres or half-mass densities, and therefore only the total mass function is presented. Similarly to Figures 6.1 and 6.2, mass-loss has caused some clusters to populate bins below the minimum CIMF mass. Since the tidal shocking mass-loss rate depends linearly on cluster mass (cf. equation 5.50), destruction of these very low-mass clusters is inefficient and consequently a large population is established.

The middle panel of Figure 6.3 shows that tidal shocking has had a much more significant affect on the Globular Cluster Velocity Distribution (GCVD). At fixed orbital energy, clusters with smaller orbital pericentres have higher radial speeds. Thus, since tidal shocking destroys clusters on orbits with small pericentres very efficiently (cf. equation 5.50), an additional affect is to reduce the local radial velocity dispersion, which is visible in the middle panel as the reduced velocity anisotropy parameter,

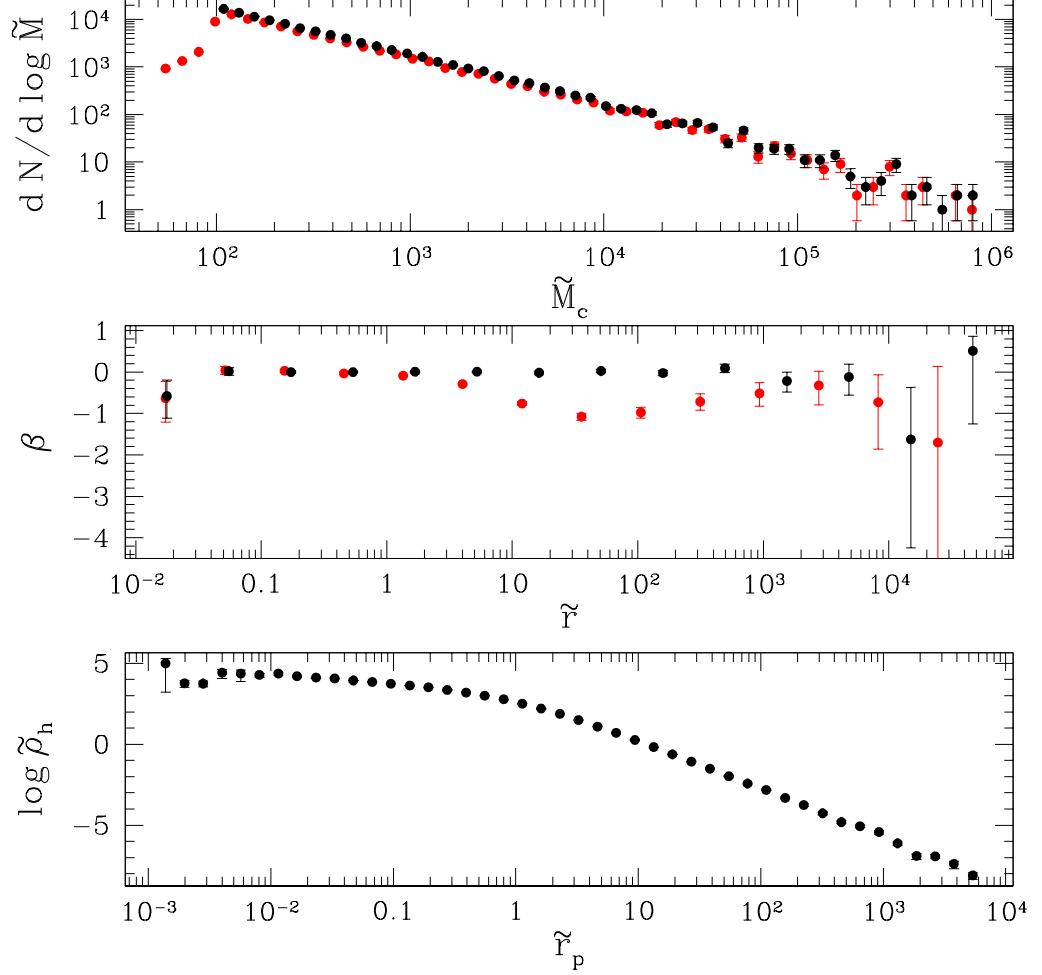


Figure 6.3: Properties of an initially isotropic GCS with a Schechter CIMF that directly traced a $\gamma = 1$ Dehnen model host galaxy. Initial half-mass densities were assigned by drawing a random King model concentration from a Gaussian with mean $\bar{c} = 2.15$ and variance $\sigma_c = 1$. Top panel: initial (black) and final (red) mass functions; middle panel: initial (black) and final (red) velocity anisotropy curves; bottom panel: half-mass density vs orbital pericentre. The only active evolutionary mechanism was tidal shocking, given by equation (5.50).

$\beta = 1 - \tilde{\sigma}_t^2 / 2\tilde{\sigma}_r^2$. Intuitively the expectation would be that β would continue to become more tangentially biased at smaller radii, however this is clearly not the case. The

reason for this is displayed in the bottom panel, showing the mean half-mass density of clusters binned by orbital pericentre. The initial half-mass density of clusters in this GCS were assigned by drawing a random King model concentration, and combining it with the tidal density at the time averaged orbital radius. Thus, since the tidal density is much higher at smaller radii, and the randomly drawn King model concentrations were independent of radius, radial dependence was imparted to the half-mass densities. The half-mass densities of clusters on orbits with small pericentres are so high that they overwhelm the tidal shocking mass-loss rate dependence on orbital pericentre, making tidal shocking very inefficient at destroying such clusters. Consequently, the velocity anisotropy at very small radii is largely unaffected. At larger radii, the depletion of radial orbits due to tidal shocking becomes weaker again (although statistical noise does disguise this to some extent), despite the lower half-mass densities of clusters with larger orbital pericentres. This is due to the very large radial periods of orbits with large orbital pericentres (the mean radial periods of clusters binned against orbital pericentre in the same way as in the bottom panel of Figure 6.3 increase from $\tilde{T}_r \simeq 10^{-1}$ at $\tilde{r}_p = 10^{-3}$ to $\tilde{T}_r \simeq 10^8$ at $\tilde{r}_p = 10^4$). Thus, much larger radial periods more than compensate for the smaller half-mass densities, and tidal shocking becomes very inefficient at destroying clusters on large orbits. This model clearly demonstrates that tidal shocking works as expected, efficiently destroying clusters on very elliptical orbits while leaving those with more circular orbits, high densities, or high radial periods largely unaffected. Consequently, the GCVD becomes tangentially biased in areas where tidal shocking operates efficiently. Again, this model is not intended to resemble real GCSs in any way, serving purely as a demonstration instead.

6.4 Evolved Stellar Evolution Models

Figure 6.4 display a GCS with a Schechter CIMF that initially directly traced a $\gamma = 1$ Dehnen model host galaxy with an isotropic velocity distribution, evolved with stellar evolution prescriptions as given in equations (5.57) and (5.58) and depicted in Figure

5.2. As discussed in section 5.2.3, stellar evolution only depends on cluster mass, and therefore options such as host galaxy, definition of effective tides, allocation of initial half-mass density, etc., are irrelevant. Consequently, as predicted all GCs have lost the same fraction of initial mass, and thus the evolved GCMF has merely been shifted to lower masses. GCs evolved with the Chernoff & Weinberg (1990) (hereafter CW) stellar evolution prescription all lost about 30.0% of their initial mass, while those evolved with the Hurley, Pols & Tout (2000) (hereafter HPT) stellar evolution prescription all lost about 34.5% of their initial mass after 13 Gyr of evolution.

As predicted in section 5.2.3 the CW prescription for m_{rm} and m_{to} (red points) causes less severe mass-loss than the HPT prescription for m_{rm} and m_{to} (green points). Note that in both cases, the probability of ejection for white dwarfs was set to 0.1. Since the integrations start at a system age of $\tilde{t}_{start} = 30$ corresponding to $m_{to} = 6.52 M_{\odot}$ in the CW case, and to $m_{to} = 9.70 M_{\odot}$ in the HPT case, the upper mass of the stellar IMF was set to $m_u = 6.52 M_{\odot}$ in both cases. Comparatively, the CW prediction for neutron stars requires a remnant mass of $m > 8 M_{\odot}$, and does not account for the possibility of black holes at all, while the HPT prediction for neutron stars requires a progenitor mass of $m > 8.19 M_{\odot}$, while black holes require a progenitor mass greater than $24.76 M_{\odot}$. Therefore, for $m_u = 6.52 M_{\odot}$ there are no progenitors massive enough to become black holes or neutron stars, and consequently do not require ejection probabilities.

The reason that the CW stellar evolution prescription causes less severe mass-loss is due to predicting a smaller m_{to} at a given \tilde{t} than the HPT stellar evolution prescription. As a result, when all stars of mass $m_{to}(\tilde{t})$ expire, the host cluster will lose mass equal to m_{to} times the number of stars at this mass (minus whatever mass may be retained in remnants). Although between the progenitor masses of $[4.7, 8] M_{\odot}$, CW predict no remnant, below this range both predictions for remnant mass are broadly similar. Since the vast majority of total cluster mass will be in low-mass stars (i.e. $m < 4.7 M_{\odot}$) where both remnant mass predictions are similar, as stars at a given \tilde{t} expire, more mass will be lost for the prescription with greater m_{to} and similar m_{rm} .

Since in most cases mass-loss per star is more severe according to the HPT stellar evolution prescription, the expectation would be that the average stellar mass

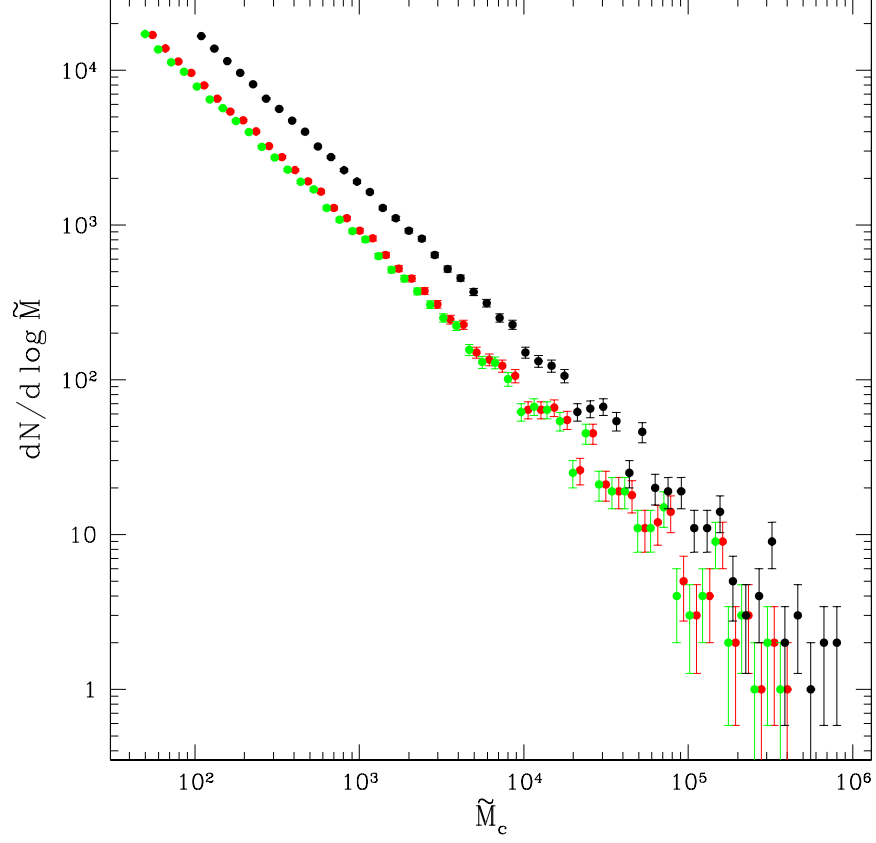


Figure 6.4: The initial (black points) and final mass functions of a Schechter CIMF that has been evolved according to the Chernoff & Weinberg (1990) (red points) and Hurley, Pols & Tout (2000) (green points) stellar evolution prescriptions for remnant mass as a function of progenitor mass, and mass of progenitor evolving off of the main-sequence as a function of time.

(including remnants) would drop more rapidly than in the CW case. Figure 6.5 displays the average stellar mass as a function of time, $\bar{m}(\tilde{t})$, for both the CW and HPT stellar evolution prescriptions, each retaining 90% of white dwarf stellar remnants.

The average stellar mass according to the HPT prescription (green curve) is initially constant because the upper mass of the stellar IMF was set to $m_u = 6.52 M_\odot$, which with this prescription does not expire until $\tilde{t} = 61.5$ (cf. Figure 5.2). As pre-

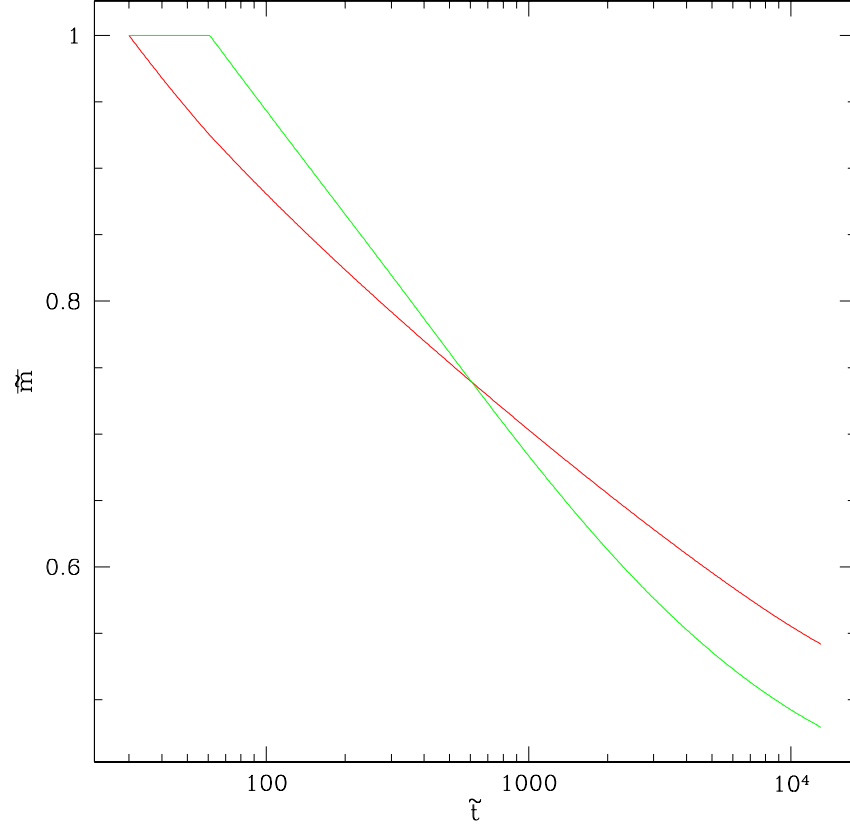


Figure 6.5: The average stellar mass including remnants as a function of time given by equation (5.55), according to the remnant mass as a function of progenitor mass (equations 5.57 and 5.58) and progenitor mass evolving off of the main-sequence as a function of time (Figure 5.2) prescriptions of Chernoff & Weinberg (1990) (red) and Hurley, Pols & Tout (2000) (green). In both cases, 90% of white dwarf stellar remnants were retained.

dicted, the average stellar mass does decrease more rapidly (once stars begin expiring) in the HPT case, dropping below the value as predicted according to the CW prescription at around $\tilde{t} = 600$. As described in section 5.2.3, the assumption is made that other mass-loss mechanisms do not preferentially target stars of any particular mass when destroying clusters, and thus do not affect the average stellar mass. Consequently, $\bar{m}(\tilde{t})$ may be combined with current cluster mass, \widetilde{M}_c , to obtain the current number of stars

in the cluster, $N(\tilde{t})$, regardless of which destruction mechanisms are active. If this treatment were applied to a cluster that had very low (or even zero) rate of mass-loss, then $N(\tilde{t})$ would increase with time. In order to prevent this unphysical behaviour, \bar{m} is only allowed to vary with time when stellar evolution is enabled, in which case all clusters will always be losing at least sufficient mass to guarantee that $N(\tilde{t})$ monotonically decreases. Some evaporation prescriptions, in addition to the half-mass density evolution prescription depend on \bar{m} (either directly or through N), and thus being able to simulate a changing average stellar mass adds an extra degree of realism to the simulations. Realistically, mass-loss mechanisms such as evaporation cause the preferential loss of low-mass stars in mass segregated clusters, which causes the average stellar mass to decrease more gradually, and at late times when the rate of decrease due to stellar evolution becomes shallow can even cause the average stellar mass to begin to increase again (e.g. Lamers, Baumgardt & Gieles 2010). However accounting for this affect requires treating the stellar population in each cluster individually, which obviously greatly adds to the computational expense. Consequently, this was not attempted (see also section 5.2.3). These models clearly demonstrate that stellar evolution operates as expected, with the evolved mass function retaining its original shape while shifting to a lower mass. Moreover, the different remnant mass and main-sequence turn-off mass functions cause expected and well understood behaviour in terms of the fraction of initial mass lost, and the rate of change of average stellar mass.

6.5 Evolved Dynamical Friction Models

Although it has long been established that dynamical friction plays a very minor role in shaping the GCMF (only very massive clusters on small orbits are significantly affected, see section 1.4.2), less attention has been paid to the role of dynamical friction in shaping the GCVD. Figure 6.6 displays the change in radial (red) and tangential (black) velocity dispersions after 13 Gyr of dynamical friction as a function of galactocentric radius. The initial GCS was isotropic, directly traced a $\gamma = 1$ Dehnen model

host galaxy, and was described by a Schechter CIMF. The final velocity dispersions were obtained by integrating the equations of motion (equations 5.9 and 5.10) for each GC including the effects of dynamical friction given by equation (5.61), as described in section 5.1, such that the orbital energies and angular momenta of orbits decrease with time. As a result, tracers end up on different orbits (with potentially very different orbital eccentricities and radii, and thus very different radial and tangential speeds). The velocity dispersions were then calculated in the normal way by binning clusters radially, and taking the standard deviations of the radial, azimuthal, and polar velocities ($\tilde{\sigma}_r, \tilde{\sigma}_\psi, \tilde{\sigma}_\theta$), and calculating the tangential velocity dispersion according to $\tilde{\sigma}_t^2 = \tilde{\sigma}_\psi^2 + \tilde{\sigma}_\theta^2$. Due to the long run-time of integrating decaying orbits, only the innermost 10^4 tracers out of the possible 10^5 were utilised, so as to obtain the most pronounced effect for the smallest computational effort.

Figure 6.6 demonstrates that dynamical friction has had several effects on the GCS. For example, clusters on very small orbits have been decayed to the point where they passed within $\tilde{r}_{min} = 10^{-4}$ and were destroyed, hence the smallest radius bin in this snapshot is at $\tilde{r} \simeq 0.7$. At larger radii, the only significant effect dynamical friction has had is to alter the shape of GC orbits rather than destroy them, as expected. At all radii, where radial velocity dispersions are decreased tangential velocity dispersions are increased, and vice-versa. An increasing velocity dispersion means that the spread in velocities at that radius is increasing, while a decreasing velocity dispersion means that the spread in velocities is decreasing. Thus, if the majority of particles at a given radius were on elliptical orbits that pass through that radius at high speed, then the majority of tangential speeds would be low, and consequently a small tangential velocity dispersion would be expected. Alternatively, if many particles had their pericentres at that radius, then their tangential speeds would be at a maximum while their radial speeds would be at a minimum. In this case, a higher tangential velocity dispersion with a lower radial velocity dispersion would be expected. Another option is that the majority of particles at a given radius have orbital apocentres there, in which case both their radial and tangential speeds will be at minimum. Thus the radial velocity dispersion would be small (since most radial speeds will be zero), while the tangential

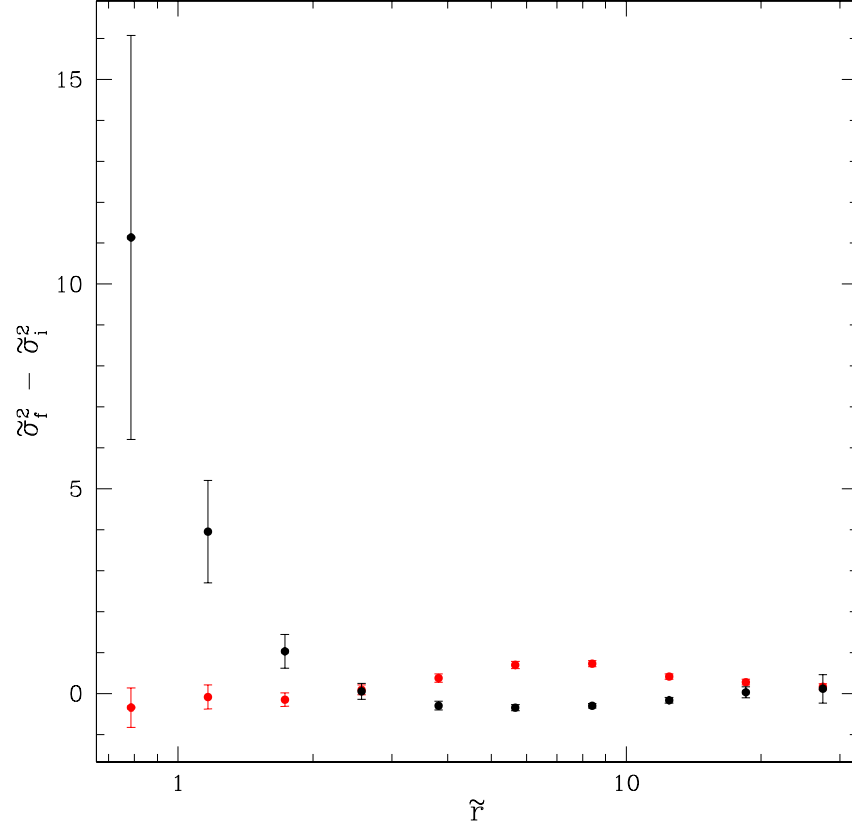


Figure 6.6: Change in tangential (black points) and radial (red points) velocity dispersions due to dynamical friction as functions of galactocentric radius. Initially the GCS was isotropic, directly traced a $\gamma = 1$ Dehnen model host galaxy, and was described by a Schechter CIMF. The final velocity dispersions were obtained by integrating the orbits of each GC including the effects of dynamical friction, given by equation (5.61).

velocity dispersion would be slightly higher (as the tangential speed will be small but non-zero, depending on the angular momentum $\tilde{v}_t = \tilde{L}/\tilde{r}_a$). The final option is that the majority of particles at a given radius are on circular orbits, in which case the majority of radial speeds will be zero, while the majority of tangential speeds will be the circular speed at that radius. Thus, both the radial and tangential velocity dispersions would be small.

At smaller radii, the velocity anisotropy appears to have become tangentially biased, with tangential velocity dispersions having been systematically increased strongly, while the radial velocity dispersions were decreased slightly. This would intuitively make sense, since orbits with pericentres at smaller radii would be strongly affected and eventually destroyed by dynamical friction. Therefore, the innermost pericentres of surviving clusters are around $\tilde{r} \simeq 0.7$, and since at pericentre radial speeds are lowest while tangential speeds are highest, a small radial velocity dispersion and a large tangential velocity dispersion would be expected. Further out at around $\tilde{r} \simeq 2$, the tangential velocity dispersions have been decreased slightly while the radial velocity dispersions have been increased, indicating either that the orbits that already passed through these radii have been made more elliptical, or higher energy orbits have been decayed such that they now pass through these radii also. Either way the velocity anisotropy has become more radially biased. At radii $\tilde{r} \gtrsim 10$, the difference between the initial and final velocity dispersions tend to zero, indicating that dynamical friction is too weak to have much impact on the GCVD beyond $\tilde{r} \gtrsim 30$.

In conclusion, even though dynamical friction is usually neglected due to having an insignificant effect on the GCMF, it is capable of having a significant impact on the innermost regions of the GCVD. The impacts are likely to be greatest in radially anisotropic GCSs, since dynamical friction has the greatest effect on small pericentre orbits (e.g. van den Bosch et al. 1999). Therefore, as radial anisotropy is expected to be necessary to produce a radially invariant GCMF, dynamical friction may be important for reproducing a GCVD consistent with observations. On the other hand, it is possible that effects such as tidal shocking could have a much stronger effect on the GCVD, in which case dynamical friction would be second order. In any event, this model demonstrates that dynamical friction is operating in a way that is intuitively expected.

6.6 Evolved GCFP Models

Figure 6.7 displays the initial (black) and evolved (red) GCFP of an initially isotropic GCS that directly traced a $k = 2.5$ polytrope host galaxy. The mass function of the GCS was initially described by a Gaussian CIMF, with $\widetilde{M}_c = 10^5$ and $\sigma_{\widetilde{M}_c} = 1$ (cf. equation 3.64). The half-mass densities of tracers were assigned according to the mass-radius relation $\tilde{r}_h = A\widetilde{M}_c^B \Rightarrow \tilde{\rho}_h = \frac{\tilde{m}_0}{M_0} \frac{3\widetilde{M}_c^{1-3B}}{2P_0A^3}$, with $A = 2.75 \times 10^{-4}$ and $B = 0.615$, depicted as the solid black line. Subsequent evolution was caused by stellar evolution under the Hurley, Pols & Tout (2000) prescription for m_{rm} and m_{to} , and half-mass density evolution as described by maintained virial equilibrium described in equation (5.105). The upper and lower mass limits on the stellar IMF were $m_u = 6.52 M_\odot$ and $m_l = 0.501 M_\odot$, and 90% of white dwarf remnants forming upon progenitor expiry were retained. Tidal densities were assumed to operate from the orbital pericentre, i.e. $\tilde{r}_{ev} = \tilde{r}_p$, and the galaxy mass-scale was assumed to be $\frac{\tilde{m}_0}{M_0} = 10^{-9}$.

The combination of these two prescriptions has produced some dramatic results, with the initial relation between half-mass radius and mass completely erased. The reason that the initial one-to-one relation between half-mass radius and cluster mass was able to become such a disperse cloud in \tilde{r}_h vs \widetilde{M}_c is due to the half-mass density evolution dependence on tidal density. With the mass-radius relation $\tilde{r}_h = A\widetilde{M}_c^B$, the initial value of κ (the parameter determining how strongly the half-mass density evolution responds to mass-loss and average stellar mass evolution) depends on mass very weakly, with $\kappa_i \propto \widetilde{M}_c^{(15B-11)/24}$ (ignoring the Coulomb logarithm, cf. equation 5.106). For $B = 0.615$, this is $\kappa_i \propto \widetilde{M}_c^{-0.074}$, and the only other parameter that κ depends on is $\tilde{\rho}_t$. Since the only mass-loss mechanism enabled in this simulation was stellar evolution, clusters that began with the same mass would also continue to have the same mass at all later times (all clusters lost 34.5% of their initial mass after 13 Gyr). Therefore for fixed tidal density, clusters of any mass will end up increasing their half-mass densities by roughly the same factor, and thus the bounding lower limit is essentially just where clusters with the smallest allowed orbital pericentre ($\tilde{r}_{min} = 10^{-4}$) and thus the highest tidal density have increased in density by the maximum

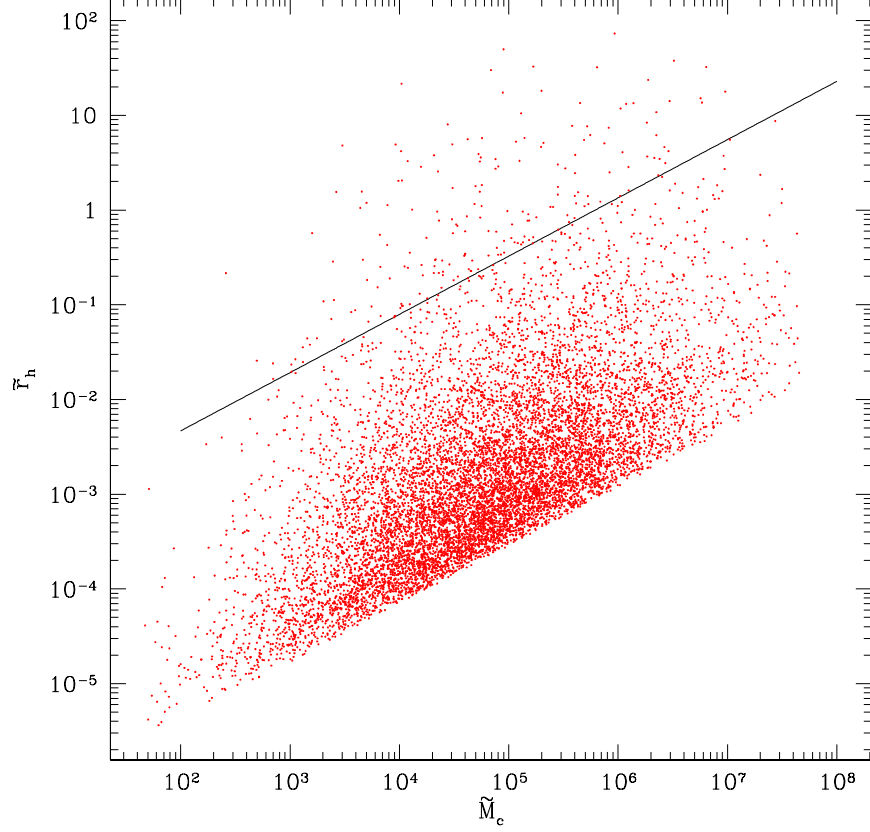


Figure 6.7: Initial and final GCFP in terms of half-mass radius and mass. Initially the GCS was isotropic, directly traced a $k = 2.5$ polytrope host galaxy, and had a Gaussian CIMF. Initial cluster half-mass densities were assigned according to the mass-radius relation $\tilde{r}_h = A \tilde{M}_c^B$, depicted as the solid black line. Evolution of the GCS was caused by stellar evolution according to the Hurley, Pols & Tout (2000) prescription, and half-mass density evolution described by equation (5.105).

amount possible, which is basically the same for all initial masses. Consequently, the lower bounding limit is the initial mass-radius relation renormalised to lower radii (modulo the tiny mass dependence in κ_i). Of course, not all clusters have the minimum allowed orbital pericentre, in which case they will have smaller tidal densities, and consequently smaller κ_i . Thus, over 13 Gyr they increase their density by a smaller amount, depending on their orbital pericentres. Therefore, the vertical spread in half-

mass radii in Figure 6.7 is a consequence of the orbital distribution in the GCS.

The fact that this prescription for half-mass density evolution effectively erases the initial mass-radius relation is encouraging, since observed GCFPs display a lack of any correlation between mass and radius (e.g. Harris 1996). When combined with a mass-loss prescription that efficiently destroys high half-mass density clusters (such as for example the evaporation prescription given by equation 5.16), this half-mass density evolution prescription combined with the initial mass-radius relation $\tilde{r}_h = A\tilde{M}_c^B$ could potentially recover the GCFP as observed in extant galaxies. Furthermore, by increasing cluster densities in response to mass-loss during evolution, the mass-loss rate due to half-mass density dependent evaporation would be accelerated. This would assist with moving the evolved GCMF peak to higher masses in Figure 6.1. However, since higher tidal densities cause more rapid increase of half-mass densities, this could impart radial dependence into the evolved GCMF. This could in turn be combated with radially anisotropic initial GCSs as made possible with the quasi-separable distribution functions.

The methods and prescriptions presented in Chapter 5 for evolving simulated GCSs as produced by the methods presented in Chapter 3 and demonstrated in Chapter 4 have been applied to several of these simulated initial GCSs to produce and present evolved simulated GCSs using each of the available evolutionary mechanisms. By comparing the initial and final properties of the constituent tracers, the effects of these mechanisms have been quantitatively and qualitatively analysed, explained, and compared to what was intuitively expected. In conclusion, the evolution procedure and associated mechanisms appears to be working as expected, and are ready to begin thoroughly exploring the vast parameter space available to these models. Potential applications for these combined initial conditions and evolutionary code are discussed in the final Chapter.

7 Summary and Discussion

7.1 Summary

In Chapter 1 the broad picture of GC formation and evolution was introduced. The properties and physics of individual GCs such as tidal densities and relaxation times were described and derived, in addition to the virial theorem (see Appendix A). Next the ensemble properties of GCSs, such as the Globular Cluster Fundamental Plane (GCFP), Globular Cluster Velocity Distribution (GCVD), and the Globular Cluster Mass Function (GCMF) were introduced. The observed properties of these aspects of the GCS were discussed, such as the lack of any detectable correlation between GC mass and half-mass radius in the GCFP, and how observations of GCVDs are consistent with isotropy, though projection effects make this somewhat uncertain. Particular emphasis was placed on the GCMF as it is the most easily observed aspect of GCSs, with the discussion moving on to how it is observed to have the same log-normal shape and peak mass that are observed to be invariant with galactocentric distance in all galaxies. The focus then moved on to how new observations of starburst systems made available with the launch of the HST lead to the exploration of the idea of GCMFs having begun with a very different shape, namely a power-law Cluster Initial Mass Function (CIMF) that continues to rise towards low masses. Subsequent evolution due to dynamical effects over a Hubble time then modified this CIMF through destruction of GCs to produce GCMFs as seen today. In order to accomplish this, these dynamical effects would have to preferentially destroy low-mass clusters, while leaving the high-mass end of the GCMF largely untouched. The mechanisms responsible for the destruction of GCs were derived and any sources of significant uncertainty discussed in detail. Each mechanism was then analysed in terms of how they affect GCs, and how they are likely to affect the GCMF as a result. These mechanisms are evaporation, tidal shocking, stellar evolution, dynamical friction, and core collapse. Arguments from the literature asserting which of these mechanisms are preferred were presented, with some arguing

for a virtually instantaneous change from a power-law CIMF to a log-normal GCMF, while others contend that a gradual change over a Hubble time is more likely. Various data such as the mass function of an intermediate age GCS (Goudfrooij 2012), the Milky Way GCMF binned by cluster half-mass densities, and the GCFP overlaid with a survival triangle (Gnedin & Ostriker 1997) are presented, each indicating that evaporation is primarily responsible for shaping the GCMF. The attempts of several authors to simulate the evolution of a GCMF from a power-law CIMF are discussed, and the recurring problem of radially dependent evolved GCMFs contrary to observations highlighted. Several authors attempted to erase the radial dependence of their evolved GCMFs by introducing radial velocity anisotropy into their models, but none could get models fully consistent with observations. The review then ends on the work of Jordán et al. (2007) and McLaughlin & Fall (2008), who devised a method of simulating GCMF evolution from half-mass densities rather than tidal densities, and consequently avoided building radial dependence into their evolved GCMFs, obtaining models which match observations very well.

Chapter 2 detailed the application of the same half-mass density-dependent models as McLaughlin & Fall (2008) to 22 early type Virgo galaxies. The Chapter begins by addressing the more disputed model assumptions, namely that the mass-loss rate is assumed to scale with half-mass density rather than tidal density, that half-light density is a good representation of half-mass density, and that Virgo GCs are in fact tidally limited. This is then followed by a detailed discussion of the data, including observational details such as observational limits, and how cuts were applied to the data. Furthermore, data incompleteness is discussed and accounted for using completeness tables created by Jordán et al. (2009), and mass-to-light ratios are described and used to convert between cluster luminosities and masses. A derivation of the half-mass density dependent models is then presented, with definitions such as the proportionality constant between mass-loss rate and half-mass density (C) given, in addition to a detailed discussion of the assumptions such as choice of CIMF, constant rate of evaporation mass-loss, etc. This is then followed by the results of application of these models to the least, intermediate, and most dense, in addition to the smallest, intermediate,

and largest galactocentric distance GCs of 22 early type Virgo galaxies. The effects on C of changing model parameters such as the Schechter exponential cut-off mass M_{sch} , or the power-law component gradient β_{MF} , were described, and it was concluded that C does not vary strongly. The results are then analysed in terms of how the evolved GCMF model turnovers vary with the median half-mass densities and projected galactocentric positions of the constituent GCs, with the expected dependence on half-mass density, yet invariance with galactocentric distance very apparent. This was explained in terms of a lack of correlation of half-mass densities with galactocentric distances, the same as is observed in the Milky Way (McLaughlin & Fall 2008), M104 (Chandar, Fall & McLaughlin 2007), and NGC1316 (Goudfrooij 2012). In conclusion, the very simple half-mass density-dependent models fit the GCMFs of the Virgo galaxies very well, with the exception of the two most massive galaxies. This is attributed to the complex merger histories that these galaxies must have, and other dynamical effects such as tidal shocking, dynamical friction, etc. Additionally, the value of C obtained was also in good agreement with general theory, predicting that on average GCs in Virgo should undergo about another 13 relaxation times before being completely destroyed (e.g., Hénon 1961; Spitzer & Chevalier 1973; Spitzer & Shull 1975; Fall & Rees 1977; Caputo & Castellani 1984; Spitzer 1987; Aguilar, Hut & Ostriker 1988; Chernoff & Weinberg 1990; Gnedin & Ostriker 1997; Murali & Weinberg 1997; Gnedin, Lee & Ostriker 1999; Jordán et al. 2007; Chandar, Fall & McLaughlin 2007; McLaughlin & Fall 2008; Goudfrooij 2012). However, the fact that these models had difficulty fitting the two most massive Virgo galaxies indicates that there are dynamical effects other than evaporation that are important for shaping the GCMF. Moreover, by utilising the lack of correlation between half-mass densities and galactocentric position, these models successfully reproduce GCMFs that are radially invariant. However, they do not explain why this lack of correlation exists.

This then led to the development of considerably more sophisticated models in Chapter 3, allowing the creation of initial GCSs with controllable spatial distributions, velocity distributions, CIMFs, and host galaxies. These models utilise quasi-separable distribution functions, requiring the host galaxy density profile, GCS spatial profile,

and the circularity function (a function controlling the velocity anisotropy profile of the GCS) to be pre-defined. Firstly, dimensionless notation is introduced to allow any models to be easily rescaled for fitting to observations. The isotropic distribution functions and density profiles of polytropes, King models, Dehnen models, and power-laws are given, followed by an in-depth discussion of their properties, including the range of model parameter they are defined over (i.e. $1 < k \leq 2.5$, $W_0 > 0$, and $0 \leq \gamma < 3$). A description of the procedure for solving for host galaxy properties such as gravitational potential, circular speeds, isotropic velocity dispersion, etc., is then given, involving the Poisson and Jeans equations. The pre-defined spatial profile by definition is equal to the integral over the distribution function, which by the definition of quasi-separable distribution functions is equal to the product of a function of orbital energy and the pre-defined function of orbital circularity. Consequently the only unknown is the function of orbital energy, and to solve for this, the integral must be inverted. This integral equation is known as the Volterra equation of the first kind, and a general method for its solution is given in Appendix B. Several examples for possible circularity functions are given, and their likely affect on the initial GCVD discussed, and additionally the circularity function for constant anisotropy is analytically solved for (see Appendix C). Once the full distribution function corresponding to the required spatial profile, velocity distribution, and host galaxy is known, it may be used to solve for properties of the GCS such as radial and tangential velocity dispersions, and differential energy distribution (see Appendix D). Moreover, the procedure for using Monto Carlo sampling to generate a population of tracers adhering to the pre-defined spatial profile, velocity distribution, and CIMF is described. This procedure is used to obtain initial masses, initial galactocentric radii, orbital energies, and orbital angular momenta. Once these are known, other properties such as orbital pericentre and apocentre, radial, azimuthal and polar velocities, etc., immediately follow. Furthermore, the process by which radial and azimuthal periods can be obtained is also described, requiring the solution of a second order differential equation for orbital radius as a function of orbital azimuth. All of this physics and numerics has been compiled into a code (see Appendix E), allowing the simulation of initial GCSs in a much more general way than has been previously

attempted.

Chapter 4 then demonstrates the application of the code given in Appendix E to create a number of GCSs with a variety of host galaxies, spatial distributions, velocity distributions, and CIMFs. These GCSs are presented in terms of spatial profiles, differential energy distributions, and average radial, azimuthal and polar velocities, anisotropy profiles, and radial and tangential velocity dispersions. Furthermore, a test is presented along with every GCS to show that the spatial distribution recovered by integrating the distribution function matches the original pre-defined spatial distribution, and that therefore the solution of the distribution function via the method in Appendix B is operating as desired. Moreover, the spatial distributions, differential energy distributions, velocity dispersions, and velocity anisotropy profiles are overlayed with the numerically solved curves (the spatial distribution is additionally overlayed with the original pre-defined spatial distribution). Additionally, the χ^2_ν values measuring how well the binned sampled data match the overlayed numerical curves are presented in each case. All of these statistics and tests indicate that the initial conditions code is performing well, reliably producing the desired GCSs. Moreover, the code continues to produce the desired GCSs to a high standard even as more complicated GCSs are required, such as spatial distributions no longer directly tracing the host galaxy, increasingly complex circularity functions, and with non-zero \tilde{L}_0 . This Chapter is then concluded by exploring the effects of different circularity functions and non-zero \tilde{L}_0 on the velocity anisotropy profile. The circularity functions used were the examples given in Chapter 3, and each was explored with several values of a (and b where appropriate). The changes to the velocity anisotropy profile in response to changes in the circularity function are generally well understood, and it was noted that a reasonably accurate prediction can be made based on the average gradient of the circularity function, with increasingly negative average gradients producing radial anisotropy, near-zero average gradients producing very slight anisotropy, and increasingly positive average gradients producing tangential anisotropy. Furthermore, larger values of \tilde{L}_0 were found to force isotropy in the core, pushing the effects of the circularity function to higher radii such that \tilde{L}_0 has very little impact on the velocity anisotropy asymptotically.

Chapter 5 details the physics of the time evolution of GCs, beginning with the equations of motion. These equations are derived without making the standard assumption that the rate of change of orbital energy and angular momentum are zero, allowing the insertion of a prescription predicting orbital decay. Following this, a thorough literature review was conducted, and a large number of evaporation prescriptions collected, each with a short discussion about what their basic assumptions are and how they were derived. Each of these evaporation prescriptions is then compiled into a table and divided into components based on mass dependence, density dependence, orbital dependence, and what host galaxy they were derived for. A detailed discussion is then given comparing the prescriptions, and predicting how they would affect the GCMF differently. The next mass-loss mechanism described is tidal shocking, with the only prescription that of Gnedin & Ostriker (1997). A detailed derivation is presented, including corrections for adiabatic shocks, point mass perturbers, and the time variation of the tidal field. Furthermore, predictions as to how tidal shocks would affect the GCMF are given, based on the mass-loss rate dependence on cluster properties. Following this, a full derivation of the mass-loss rate due to stellar evolution is given, in addition to the average stellar mass as a function of time. Two prescriptions are presented for the stellar remnant mass as a function of progenitor mass, and mass of progenitor star evolving off of the main-sequence as a function of time. These two prescriptions are compared, and predictions are made as to how the mass-loss rate and average stellar mass would behave in each case. The penultimate mechanism described is dynamical friction, with the only prescription being that of Chandrasekhar (1943), presented along with a discussion of potential weaknesses and how to address them, in addition to an improved representation of the Coulomb logarithm by Just & Peñarrubia (2005). Furthermore, an equation predicting whether a particle on a decaying orbit is instantaneously being placed on a more or less elliptical orbit is derived, which makes the prediction that a particle on a decaying circular orbit will only decay through a series of circular orbits if $\tilde{L}_0 = 0$; otherwise the ellipticity of the orbit must increase. Finally, the last mechanism to be discussed is the internal evolution of GCs. The only prescription for this mechanism is that of maintained virial equilibrium, which

is presented along with the full derivation, followed by a physical interpretation and predictions as to how a GC evolving according to this prescription would behave.

Finally, in Chapter 6, the code applying the evolutionary routines and prescriptions of Chapter 5 to the initial simulated GCSs of Chapters 3 and 4 is described (see Appendix F), including potential numerical pit-falls and how they were dealt with. Following this, examples are given of the application of the evolution code to produce evolved GCS models. The first set of evolved GCSs presented were evolved with evaporation prescriptions. The first of these began as a Schechter CIMF, and all initial half-mass densities were set according to the relation $\tilde{r}_h = 2.75 \times 10^{-4} \widetilde{M}_c^{0.615}$. The evaporation prescription acting on this initial GCS over 13 Gyr was that of equation (5.16), and consequently the mass-loss rate depended only on cluster half-mass density, and therefore equivalently to initial mass. The evolved GCMF produced in this case has the characteristic log-normal shape, but the turnover mass is about an order of magnitude too low. The second GCS evolved by evaporation began as a Gaussian CIMF, and had initial half-mass densities set by randomly sampling a Gaussian of King model concentrations and combining the corresponding ratio of half-mass density to tidal density with the tidal density at the time-averaged orbital radius. Consequently, since tidal densities are higher for clusters on smaller orbits, the half-mass densities had a built in radial dependence. Furthermore, the evaporation prescription used in this case was that of equation (5.20), which depends on half-mass density, the ratio of tidal density to half-mass density, and mass. Consequently this mass-loss rate is higher for smaller orbits, where clusters have higher half-mass densities. As a result, the GCMF peak mass can be seen to be slightly higher when constituent GCs are binned by orbital pericentre. In this case the evolved GCMF peak mass was at around $\widetilde{M}_{TO} \simeq 10^{5.5}$. The next mechanism demonstrated was tidal shocks, which had been acting on a GCS initially described by an isotropic velocity distribution and a Schechter CIMF, with initial half-mass densities allocated again by randomly sampling a Gaussian of King concentrations, the same as previously. In this case the total mass-loss was very slight, with no discernible change of shape to the mass function. However the velocity anisotropy clearly displayed that the central regions had become tangentially biased. Clusters on

smaller orbits had been protected by their very large half-mass densities, while those far out had been protected by their very large radial periods. The subsequent mechanism demonstrated was stellar evolution, which as expected had reduced the initial mass of every cluster by the same fraction, and consequently the evolved GCMF was just shifted to a lower mass. Additionally, the average stellar mass as a function of time was displayed for both available prescriptions, behaving as predicted in both cases. Dynamical friction was demonstrated next, with a plot displaying how the radial and tangential velocity dispersions had changed over 13 Gyr. The changes were consistent with the idea that clusters on orbits with very small pericentres had been destroyed or circularised, such that the smallest pericentres were now typically further out, and consequently the tangential velocity dispersion had increased while the radial velocity dispersion had decreased at this radius. The final demonstration was that of the half-mass density evolution and stellar evolution mechanisms, in a GCS initially described by a Gaussian CIMF, and with initial mass-radius relation of $\tilde{r}_h = 2.75 \times 10^{-4} \tilde{M}_c^{0.615}$. A Hubble time of evolution had caused the initial mass-radius relation to be completely erased, with the only trace of its existence being the lower bound to the final \tilde{r}_h vs \tilde{M}_c of GCs. The large vertical spread in points was explained as being a direct consequence of the orbital distribution of the GCS, as higher tidal densities correspond to more rapid changes in half-mass densities, and the initial mass-radius relation caused the rate of change of half-mass density to be very insensitive to initial mass/half-mass density.

In summary, the physics of GC evolution and destruction have been either presented or derived and discussed, with a focus on how the evolution of individual GCs affects the properties of the ensemble GCS as a whole, and in particular the mass function. These evolutionary mechanisms were analysed in terms of how they would affect the GCMF under the assumption that it began as a power-law CIMF. It was concluded that evaporation is most likely primarily responsible for shaping the GCMF, and models built with these assumptions are applied to 22 Virgo galaxies. An estimate is obtained for the coefficient on evaporation mass-loss rate that is in good agreement with other similar studies and with evaporation theory in general, however these mod-

els are found to struggle to fit the GCMFs of very massive galaxies with very populous GCSs. Consequently, much more sophisticated models are derived, able to create initial GCSs with specifiable spatial distribution, velocity distribution, CIMF, and host galaxy. Furthermore, these initial models can then be evolved while applying any combination of evaporation, tidal shocking, stellar evolution, dynamical friction, and internal evolution mechanisms. All of the primary mechanisms affecting the evolution of GCs have been either derived or taken from the literature and woven together to make the treatment of GC evolution as general and realistic as possible. Consequently, this powerful evolution routine should be well poised to make significant contributions to the field of Globular Cluster studies, for example answering long standing questions surrounding GCS evolution, such as:

- Why the GCMF is radially invariant, and what role velocity anisotropy plays.
- Whether the GCMF evolved from a CIMF analogous to power-law CIMFs as observed in sites of heightened star formation such as galactic mergers.
- Whether GCs are the survivors of an initially much larger population, and how large that population may have been.
- Whether GCs began with an initial mass-radius relation, and what it may have been.
- Why GC half-mass radii and masses are uncorrelated.

7.2 Future Work

The next step with these initial condition and evolution routines is to begin to thoroughly explore the vast parameter space in a systematic way. For example, these models were initially developed for application to the most massive Virgo galaxies, since the simple single parameter models produced unsatisfactory fits. However given

that the single parameter models were able to fit the remaining 20 Virgo galaxies, a large amount of additional physics may not be necessary to reconcile the fits with these massive galaxies. Due to the formulation of the single parameter models, the slope of the Schechter CIMF, β_{MF} , is degenerate with C , and thus varying β_{MF} in an attempt to achieve better fits would have been pointless. However, these more sophisticated routines allow other parameters and properties to vary, which could break this degeneracy. For example, the evolved GCMF in Figure 6.1 is remarkably close to the appearance of actual GCMFs considering it was intended purely as a demonstration that the evolution routine works. The input physics and parameters need to be adjusted such that the turnover of the resulting GCMF is about an order of magnitude greater. It is entirely possible that decreasing β_{MF} slightly or increasing C by a factor of two would address this. Alternatively, this could be achieved by altering the initial mass-radius relation in favour of slightly higher densities, as this would correspond to higher rates of mass-loss. Decreasing A would increase the initial density of all clusters by the same factor, and consequently the mass lost from every cluster would be multiplied by the square root of this factor. Thus, the more severe mass-loss would erode more of the GCMF, and the resulting turnover mass would be higher. On the other hand, decreasing B would preferentially increase the initial densities of more massive clusters, and thus mass-loss rates would be higher for all clusters, but especially for more massive clusters. Consequently the shape of the resulting GCMF would be modified, with a larger fraction of massive clusters migrating to lower masses, which could actually lower the turnover mass if B was decreased enough.

Other aspects of the GCS also need to be considered, such as the GCFP. Since half-mass densities are assumed to be constant, changing either A or B significantly would have a large effect on the GCFP of the evolved GCS, such that surviving clusters have half-mass densities orders of magnitude away from what is actually observed. Furthermore, an even more fundamental problem is that by assuming any mass-radius relation combined with constant half-mass densities and a mass-loss rate that scales with half-mass density, a clear one-to-one relation will always be visible in any surviving GC populations. Such a feature is not seen in real GCFPs. Scatter in $\tilde{\rho}_h$ vs \tilde{M}_c

could be achieved by using an evaporation prescription that also depends on tidal density for example, since the orbital distribution of clusters would then guarantee a spread in the rates of mass-loss. Alternatively, internal evolution could be utilised, since Figure 6.7 demonstrates that internal evolution is very efficient at erasing the initial mass-radius relation, again due to the orbital distribution. However in both of these alternatives, dependence on orbital distribution automatically implies radial dependence in the mass-loss rates, and therefore also in the resulting GCMF. How strong this radial dependence would be is currently unclear, but could potentially be combated with radially anisotropic models. Furthermore, since internal evolution increases the half-mass densities of many clusters, the evaporation mass-loss rate will accelerate. Consequently the turnover mass of the resulting GCMF would be at higher masses without needing to alter either A or B . Internal evolution will also cause the half-mass density of many clusters to decrease, which would therefore be very resistant to evaporation mass-loss and as a result would accumulate. Again this is not observed in GCSs (though if they were very distended they would be hard to detect). However tidal shocking would be very efficient at destroying such low half-mass density clusters and thus could resolve this potential issue anyway.

Overall, there are several possible paths that could be taken to produce simulated evolved GCMFs resembling observed GCSs. It is not difficult to imagine a scenario where introducing additional mechanisms would produce unwanted side-effects, which could be countered by another mechanism that brings more unwanted side-effects of its own, etc., and thus a minimalistic approach should be taken. Careful exploration of the parameter space one mechanism at a time to find the minimum complexity required to produce a well fitting model will be the main goal. Once a combination of mechanisms and parameters providing evolved GCSs that match well to observations have been found, these models should be projected and applied to the GCSs of other galaxies, and in particular the Virgo galaxies so as to use these sophisticated models for their originally intended purpose. Given that many of the galaxies these models would be applied to are much more massive, in addition to having very different structures (i.e. ellipticals as opposed to spirals), it would not come as much of a surprise if the set of

mechanisms and parameters providing models that fit well in the Milky Way require some modification in order to provide good fits to Virgo galaxies as well. Thus, some exploration of the effects of changing mechanisms and parameters on the projected models may be necessary.

Furthermore, there is the potential to add further improvements to the code, such as relaxing the assumption of spherical symmetry to allow the use of axisymmetric host galaxy potentials and/or GCS spatial distributions. This would allow more direct comparisons to GCSs in disk galaxies, in addition to investigation of how much difference the presence of a galactic disk would have on GC evolution. An obvious place to start would be with tidal shocks, since in this case disk crossings would need to be considered. Additionally, there is the possibility of relaxing the assumption of time-independent potentials to more accurately reflect the hierarchical nature of galaxy evolution. This would likely be rather a complicated undertaking without N-body simulations, but could potentially be simulated in a statistical sense, with random numbers used to reflect GC orbit migration during violent relaxation, in addition to the number of accreted GCs and the scale and inclination of the merger. This would allow investigation into how significant a role orbital mixing has on GC evolution, and could be the reason for so much discrepancy in tidal densities of clusters obtained by King model fits compared to theoretical predictions of tidal density based on the orbit of the cluster in question. For example, a cluster from a recently cannibalised host galaxy could have spent the majority of its lifetime in a very different environment, and could still be adjusting to its new environment. Thus the observed properties of such GCs would reflect the previous rather than the current environment.

Additionally, the prescriptions for stellar evolution and internal evolution could be improved. For example, mass-loss could be treated in a way that conveys the probability that the lost stars were of a certain mass, thus making the average stellar mass reflect effects such as the preferential loss of low-mass stars. For example, the calculations for evolving stellar populations could be improved by treating for radial dependence of the stellar mass function, reflecting ongoing mass segregation through a statistical treatment of local density-dependent stellar encounters. There is also the

possibility of accounting for primordial binaries, density-dependent binary formation, and binary reheating in the internal evolution prescription, which would slow down the rate at which clusters increase in density. This could potentially have a significant impact on how any other mechanisms depending on half-mass density behave. An even more detailed alternative would be to combine the stellar population evolution and internal density evolution prescriptions. Such a model would treat for the local density-dependent probability of stellar encounters, and the local stellar mass function dependent probability of what class and mass the components of the encounter would be (i.e., main-sequence, white dwarf, neutron star, or black hole). This would enable calculation of the probability of whether the components of the encounter migrated inwards, outwards, formed binaries, or coalesced. Consequently it would be possible to keep track of the evolving stellar mass function at different radii throughout a cluster, including the relative number and position of binaries with different class and mass components, or even the growth of an intermediate mass black hole. Such predictions could be compared to observed X-ray emissions as an additional constraint on the models. Thus, as ever, there are many possible directions that this research could be taken in next.

In conclusion, an opportunity for great progress in Globular Cluster studies has been provided by the creation of these advanced GCS set-up and evolution routines, which could potentially resolve some long-standing questions. Once these results have been obtained, determining in which way to build upon this research will become a much clearer choice.

A Virial Theorem

The total moment of inertia for a system is just the sum of the moments of inertia of the individual N stars,

$$I_{tot} = \sum_{k=1}^N m_k |\mathbf{r}_k|^2$$

where m_k is the mass of the k 'th star, and \mathbf{r}_k is the position of the k 'th star relative to the system centre of mass. Then assuming that m_k is constant, the second time derivative of I_{tot} is given by:

$$\frac{d^2 I_{tot}}{dt^2} = 2 \sum_{k=1}^N m_k \left| \frac{d\mathbf{r}_k}{dt} \right|^2 + 2 \sum_{k=1}^N \mathbf{F}_k \cdot \mathbf{r}_k$$

The first term on the right hand side is four times the total kinetic energy, K_{tot} . The second term is the scalar product of the total vector force acting on the k 'th star due to the sum of gravitational attractions to all other particles in the system, \mathbf{F}_k , and the vector position of the star. \mathbf{F}_k can therefore be expanded into a double sum, as follows:

$$2 \sum_{k=1}^N \mathbf{F}_k \cdot \mathbf{r}_k = 2 \sum_{k=1}^N \sum_{j=1}^{k-1} \mathbf{F}_{kj} \cdot \mathbf{r}_k + 2 \sum_{k=1}^N \sum_{j=k+1}^N \mathbf{F}_{kj} \cdot \mathbf{r}_k$$

where \mathbf{F}_{kj} is the gravitational force experienced by star k due to star j , and $j = k$ has been excluded since a star is not bound to a system by its own gravity. The inner sum of the first term on the right hand side is just counting $j < k$, and the inner sum of the second term is counting $j > k$. Equivalently, the second term may be written as $k < j$, and the summation variables exchanged to give:

$$2 \sum_{k=1}^N \mathbf{F}_k \cdot \mathbf{r}_k = 2 \sum_{k=1}^N \sum_{j=1}^{k-1} \mathbf{F}_{kj} \cdot \mathbf{r}_k + 2 \sum_{j=1}^{k-1} \sum_{k=1}^N \mathbf{F}_{jk} \cdot \mathbf{r}_j$$

Then noting that $\mathbf{F}_{kj} = -\mathbf{F}_{jk}$, this may be written as:

$$2 \sum_{k=1}^N \mathbf{F}_k \cdot \mathbf{r}_k = 2 \sum_{k=1}^N \sum_{j=1}^{k-1} \mathbf{F}_{kj} \cdot \mathbf{r}_{kj}$$

where $\mathbf{r}_{kj} = (\mathbf{r}_k - \mathbf{r}_j)$. By definition, $\mathbf{F}_{kj} = -\nabla\Phi_{kj} = -\frac{d\Phi_{kj}}{dr}\mathbf{r}_{kj}/|\mathbf{r}_{kj}|$, where Φ_{kj} is the gravitational potential of star j as experienced by star k . Then assuming a power-law potential for the member stars, $\Phi_{kj} = A|\mathbf{r}_{kj}|^n$:

$$2 \sum_{k=1}^N \mathbf{F}_k \cdot \mathbf{r}_k = -2n \sum_{k=1}^N \sum_{j=1}^{k-1} \Phi_{kj} = -2nW_{tot}$$

where W_{tot} is the total potential energy in the system, and:

$$\frac{d^2 I_{tot}}{dt^2} = 4K_{tot} - 2nW_{tot}$$

by taking the time average of this expression, it is reasonable to expect that for many systems $\left\langle \frac{d^2 I_{tot}}{dt^2} \right\rangle \sim 0$, and thus finally:

$$2 \langle K_{tot} \rangle - n \langle W_{tot} \rangle = 0$$

B Trapezoid Quadrature

By definition of the distribution function:

$$\tilde{\rho}(\tilde{\Phi}) = \frac{4\pi}{\tilde{r}(\tilde{\Phi})} \int_{\tilde{\Phi}}^{\infty} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right] \int_0^{h_{max}} \frac{hj(h) dh}{\sqrt{h_{max}^2 - h^2}}$$

where h_{max} is a function of \tilde{E} and $\tilde{\Phi}$ (equation 3.49). This can easily be written as:

$$\begin{aligned} & \frac{\tilde{\rho}(\tilde{\Phi})\tilde{r}(\tilde{\Phi})}{4\pi \ln(10)} - \int_{\log \tilde{\Phi}(\tilde{r}_f)}^{\infty} d \log \tilde{E} \tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right] \int_0^{h_{max}} \frac{hj(h) dh}{\sqrt{h_{max}^2 - h^2}} \\ &= \int_{\log \tilde{\Phi}}^{\log \tilde{\Phi}(\tilde{r}_f)} d \log \tilde{E} \tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right] \int_0^{h_{max}} \frac{hj(h) dh}{\sqrt{h_{max}^2 - h^2}} \end{aligned}$$

then denoting:

$$\begin{aligned} f(\tilde{\Phi}) &= \frac{\tilde{\rho}(\tilde{\Phi})\tilde{r}(\tilde{\Phi})}{4\pi \ln(10)} - \int_{\log \tilde{\Phi}(\tilde{r}_f)}^{\infty} d \log \tilde{E} \tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right] \int_0^{h_{max}} \frac{hj(h) dh}{\sqrt{h_{max}^2 - h^2}} \\ K(\tilde{E}, \tilde{\Phi}) &= \tilde{E} \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right] \int_0^{h_{max}} \frac{hj(h) dh}{\sqrt{h_{max}^2 - h^2}} \end{aligned}$$

and dividing the range $[\tilde{\Phi}, \tilde{\Phi}(\tilde{r}_f)]$ into $\mathcal{N} - 1$ strips:

$$f(\tilde{\Phi}_n) = \sum_{i=n}^{\mathcal{N}-1} \int_{\log \tilde{\Phi}_i}^{\log \tilde{\Phi}_{i+1}} \tilde{g}(\tilde{E}) K(\tilde{E}, \tilde{\Phi}_n) d \log \tilde{E}$$

The value of the integral of each of these strips is just the area under the curve of the integrand, which for a narrow enough strip can be well approximated by a trapezium:

$$\begin{aligned} \int_{\log \tilde{\Phi}_i}^{\log \tilde{\Phi}_{i+1}} \tilde{g}(\tilde{E}) K(\tilde{E}, \tilde{\Phi}_n) d \log \tilde{E} &\simeq h_i \tilde{g}(\tilde{\Phi}_{i+1}) K(\tilde{\Phi}_{i+1}, \tilde{\Phi}_n) \\ &+ 0.5h_i \left[\tilde{g}(\tilde{\Phi}_i) K(\tilde{\Phi}_i, \tilde{\Phi}_n) - \tilde{g}(\tilde{\Phi}_{i+1}) K(\tilde{\Phi}_{i+1}, \tilde{\Phi}_n) \right] \end{aligned}$$

where h_i is just the width of the strip; i.e. $h_i = \tilde{\Phi}_{i+1} - \tilde{\Phi}_i$. Then:

$$f(\tilde{\Phi}_n) \simeq \sum_{i=n}^{\mathcal{N}-1} 0.5h_i \left[\tilde{g}(\tilde{\Phi}_i) K(\tilde{\Phi}_i, \tilde{\Phi}_n) + \tilde{g}(\tilde{\Phi}_{i+1}) K(\tilde{\Phi}_{i+1}, \tilde{\Phi}_n) \right]$$

Writing this as two individual sums and changing the summation index of the second sum such that $i_{old} + 1 = i_{new}$, the following is obtained:

$$f(\tilde{\Phi}_n) \simeq \sum_{i=n}^{\mathcal{N}-1} 0.5h_i \tilde{g}(\tilde{\Phi}_i) K(\tilde{\Phi}_i, \tilde{\Phi}_n) + \sum_{i=n+1}^{\mathcal{N}} 0.5h_{i-1} \tilde{g}(\tilde{\Phi}_i) K(\tilde{\Phi}_i, \tilde{\Phi}_n)$$

The n 'th term of the first sum and the \mathcal{N} 'th term of the second sum can be taken outside their respective summations to get:

$$\begin{aligned} f(\tilde{\Phi}_n) \simeq & 0.5h_n \tilde{g}(\tilde{\Phi}_n) K(\tilde{\Phi}_n, \tilde{\Phi}_n) + \sum_{i=n+1}^{\mathcal{N}-1} 0.5h_i \tilde{g}(\tilde{\Phi}_i) K(\tilde{\Phi}_i, \tilde{\Phi}_n) \\ & + 0.5h_{\mathcal{N}-1} \tilde{g}(\tilde{\Phi}_{\mathcal{N}}) K(\tilde{\Phi}_{\mathcal{N}}, \tilde{\Phi}_n) + \sum_{i=n+1}^{\mathcal{N}-1} 0.5h_{i-1} \tilde{g}(\tilde{\Phi}_i) K(\tilde{\Phi}_i, \tilde{\Phi}_n) \end{aligned}$$

but $K(\tilde{\Phi}_n, \tilde{\Phi}_n) = 0$, so that finally:

$$f(\tilde{\Phi}_n) \simeq 0.5h_{\mathcal{N}-1} \tilde{g}(\tilde{\Phi}_{\mathcal{N}}) K(\tilde{\Phi}_{\mathcal{N}}, \tilde{\Phi}_n) + \sum_{i=n+1}^{\mathcal{N}-1} \frac{h_{i-1} + h_i}{2} \tilde{g}(\tilde{\Phi}_i) K(\tilde{\Phi}_i, \tilde{\Phi}_n)$$

where it is understood that the sum only gets done when $n \leq \mathcal{N} - 2$, such that $n = \mathcal{N}$ gives $f(\tilde{\Phi}_{\mathcal{N}}) \simeq 0.5h_{\mathcal{N}-1} \tilde{g}(\tilde{\Phi}_{\mathcal{N}}) K(\tilde{\Phi}_{\mathcal{N}}, \tilde{\Phi}_{\mathcal{N}}) = 0$ (due to both arguments of the kernel being equal) as required. Preceding terms can then be evaluated one by one to solve for all $\tilde{g}(\tilde{\Phi}_i)$ from $i = \mathcal{N}$ down to $i = 2$ (the $i = 1$ term is lost again due to the kernel collapsing to zero for $\tilde{E} = \tilde{\Phi}$) by taking out the $n + 1$ 'th term of the sum and making $\tilde{g}(\tilde{\Phi}_{n+1})$ the subject:

$$\begin{aligned} \tilde{g}(\tilde{\Phi}_{\mathcal{N}}) & \simeq \frac{2f(\tilde{\Phi}_{\mathcal{N}-1})}{h_{\mathcal{N}-1} K(\tilde{\Phi}_{\mathcal{N}}, \tilde{\Phi}_{\mathcal{N}-1})} \\ \tilde{g}(\tilde{\Phi}_n) & \simeq \frac{2f(\tilde{\Phi}_{n-1})}{(h_n + h_{n-1}) K(\tilde{\Phi}_n, \tilde{\Phi}_{n-1})} - \frac{h_{\mathcal{N}-1}}{h_n + h_{n-1}} \tilde{g}(\tilde{\Phi}_{\mathcal{N}}) \frac{K(\tilde{\Phi}_{\mathcal{N}}, \tilde{\Phi}_{n-1})}{K(\tilde{\Phi}_n, \tilde{\Phi}_{n-1})} \\ & \quad - \frac{\sum_{i=n+1}^{\mathcal{N}-1} (h_i + h_{i-1}) \tilde{g}(\tilde{\Phi}_i) K(\tilde{\Phi}_i, \tilde{\Phi}_{n-1})}{(h_n + h_{n-1}) K(\tilde{\Phi}_n, \tilde{\Phi}_{n-1})} \end{aligned}$$

where the index over n has been changed according to $n_{old} + 1 = n_{new}$.

C Constant β Circularity Function

Using the definitions of the anisotropy parameter, β , and the tangential and radial velocity dispersions, $\tilde{\sigma}_t^2, \tilde{\sigma}_r^2$ (equations 3.41, 3.55 and 3.54), β as a function of the circularity function $j(h)$ may be written as:

$$\beta = 1 - \frac{\int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^3 \int_0^{h_{max}} \frac{h^3 j(h) dh}{\sqrt{h_{max}^2 - h^2}}}{2 \int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^3 \int_0^{h_{max}} \sqrt{h_{max}^2 - h^2} h j(h) dh}$$

then using integration by parts on the inner integral in the numerator:

$$\begin{aligned} \int_0^{h_{max}} \frac{h^3 j(h)}{\sqrt{h_{max}^2 - h^2}} dh &= 2 \int_0^{h_{max}} \sqrt{h_{max}^2 - h^2} h j(h) dh + \int_0^{h_{max}} \sqrt{h_{max}^2 - h^2} h^2 \frac{dj}{dh} dh \\ &\quad - \left[h^2 j(h) \sqrt{h_{max}^2 - h^2} \right]_0^{h_{max}} \end{aligned}$$

under the condition that $h^2 j(h) \rightarrow 0$ as $h \rightarrow 0$, the last term dissapears. Putting this back into the expression for β and expanding:

$$\beta = - \frac{\int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^3 \int_0^{h_{max}} \sqrt{h_{max}^2 - h^2} h^2 \frac{dj}{dh} dh}{2 \int_{\tilde{\Phi}}^{\tilde{E}_{max}} d\tilde{E} \tilde{g}(\tilde{E}) \left[\tilde{L}_0 + \tilde{L}_c(\tilde{E}) \right]^3 \int_0^{h_{max}} \sqrt{h_{max}^2 - h^2} h j(h) dh}$$

then for $\beta = \beta_0 = \text{const}$, the following must hold:

$$\frac{dj}{dh} = - \frac{2j(h)\beta_0}{h}$$

and thus

$$j(h) = Ch^{-2\beta_0}$$

where C is an arbitrary constant of integration. Then recalling the condition that $h^2 j(h) \rightarrow 0$ as $h \rightarrow 0$, this result holds only for $\beta_0 < 1$.

D Differential Energy Distribution

By definition, the total number of tracers is given by:

$$N_{tot} = \int_{\text{all } r, v} d^3r d^3v f(H, L)$$

where H is the orbital energy. Then the number of objects with energies $H = E$ is given by:

$$\frac{dN}{dE} = \int_{\text{all } r, v} d^3r d^3v \delta(E - H) f(H, L)$$

where $\delta(E - H)$ is the Dirac delta function, i.e. $\delta(E \neq H) = 0$, and $\delta(E = H) = \infty$.

Putting this into dimensionless notation and writing $d^3\tilde{v} = d\tilde{v}_r d\tilde{v}_\theta d\tilde{v}_\phi$, this is:

$$\frac{dN}{d\tilde{E}} = \int_{\text{all } \tilde{r}} d^3\tilde{r} \int_{\text{all } \tilde{v}} d\tilde{v}_r d\tilde{v}_\theta d\tilde{v}_\phi \tilde{\delta}(\tilde{E} - \tilde{H}) \tilde{f}(\tilde{H}, \tilde{L})$$

As in the derivation of equation (3.50) (number density from the distribution function), using the definitions in equations (3.45) and (3.46), i.e. $\tilde{v}_\theta = \tilde{v}_t \cos \gamma$ and $\tilde{v}_\phi = \tilde{v}_t \sin \gamma$, the Jacobian gives $d\tilde{v}_\theta d\tilde{v}_\phi = \tilde{v}_t d\tilde{v}_t d\gamma$, and thus:

$$\frac{dN}{d\tilde{E}} = \int_{\text{all } \tilde{r}} d^3\tilde{r} \int_{-\infty}^{\infty} d\tilde{v}_r \int_0^{\infty} d\tilde{v}_t \tilde{v}_t \int d\gamma \tilde{\delta}(\tilde{H} - \tilde{E}) \tilde{f}(\tilde{H}, \tilde{L})$$

Assuming spherical symmetry, this then becomes:

$$\frac{dN}{d\tilde{E}} = (4\pi)^2 \int_{\tilde{r}_1}^{\tilde{r}_2} d\tilde{r} \tilde{r}^2 \int_0^{\infty} d\tilde{v}_r \int_0^{\infty} d\tilde{v}_t \tilde{v}_t \tilde{\delta}(\tilde{E} - \tilde{H}) \tilde{f}(\tilde{H}, \tilde{L})$$

where $\tilde{r}_1 = \tilde{r}_{\min, \text{trace}}$, and $\tilde{r}_2 = \min[\tilde{r}_{\max, \text{trace}}, \tilde{r}(\tilde{\Phi} = \tilde{E})]$, i.e. integrating over the full range of possible/allowed tracer radii. Then similarly to the derivation of equation (3.50) again, using the definitions in equations (3.44) and (3.48), i.e. $\tilde{v}_t = \tilde{L}/\tilde{r}$ and $\tilde{v}_r = \sqrt{2(\tilde{H} - \tilde{\Phi}) - \frac{\tilde{L}^2}{\tilde{r}^2}}$, the Jacobian gives $d\tilde{v}_r d\tilde{v}_t = \frac{1}{\tilde{r}\tilde{v}_r} d\tilde{H} d\tilde{L}$, resulting in:

$$\frac{dN}{d\tilde{E}} = (4\pi)^2 \int_{\tilde{r}_1}^{\tilde{r}_2} d\tilde{r} \tilde{r} \int_{\text{all } \tilde{H}} d\tilde{H} \tilde{\delta}(\tilde{E} - \tilde{H}) \int_0^{[2(\tilde{H} - \tilde{\Phi})]^{1/2} \tilde{r}} d\tilde{L} \frac{\frac{\tilde{L}}{\tilde{r}} \tilde{f}(\tilde{H}, \tilde{L})}{\sqrt{2(\tilde{H} - \tilde{\Phi}) - \frac{\tilde{L}^2}{\tilde{r}^2}}}$$

The integral over \tilde{H} may be evaluated to then give:

$$\frac{dN}{d\tilde{E}} = (4\pi)^2 \int_{\tilde{r}_1}^{\tilde{r}_2} d\tilde{r} \int_0^{\left[2(\tilde{E}-\tilde{\Phi})\right]^{1/2}\tilde{r}} d\tilde{L} \frac{\tilde{L}\tilde{f}(\tilde{E},\tilde{L})}{\sqrt{2(\tilde{E}-\tilde{\Phi}) - \frac{\tilde{L}^2}{\tilde{r}^2}}}$$

Then finally substituting the definitions of orbital circularity, quasi-separable distribution functions, and maximum dimensionless angular momentum (equations 3.1, 3.2 and 3.49), i.e.

$$\begin{aligned} h &= \frac{\tilde{L}}{\tilde{L}_c(\tilde{E}) + \tilde{L}_0} \\ \tilde{f}(\tilde{E}, h) &= \tilde{g}(\tilde{E})j(h) \\ h_{max} &= \frac{\sqrt{2(\tilde{E}-\tilde{\Phi})}\tilde{r}}{\tilde{L}_c(\tilde{E}) + \tilde{L}_0} \end{aligned}$$

the differential energy distribution may be written as:

$$\frac{dN}{d\log \tilde{E}} = (4\pi)^2 \ln(10) \tilde{E} \left[\tilde{L}_c(\tilde{E}) + \tilde{L}_0 \right] \tilde{g}(\tilde{E}) \int_{\tilde{r}_1}^{\tilde{r}_2} d\tilde{r} \tilde{r} \int_0^{h_{max}} \frac{hj(h) dh}{\sqrt{h_{max}^2 - h^2}}$$

E Simulated GCS set-up Code

Following is a code which when given a host galaxy density solves Poisson's equation for the gravitational potential and its derivatives. These are then used in combination with a selected tracer density profile and circularity function to solve a Volterra integral equation of the first kind for the energy dependence of the distribution function. Following this, Monte Carlo sampling is used to generate a population of tracers satisfying the selected tracer density profile and kinematic distribution as specified by the circularity function (see Chapter 3).

```

PROGRAM modelint
IMPLICIT NONE
INTEGER IMFnum, tracedatnum, gintnum, DEDN
INTEGER KMAXX, NMAX, intnum, dcmlpnts, itnum, voltN, objnum
REAL*8 logzero
PARAMETER (KMAXX=10000, NMAX=50, intnum=1000, dcmlpnts=1, DEDN=1000)
PARAMETER (itnum=1000, voltN=1000, objnum=200000)
PARAMETER (tracedatnum=1000, IMFnum=2000, logzero=-1.d2)
PARAMETER (gintnum=1000)
INTEGER nvar, kmax, kmax2, kount, kount2, i, nok, nbad, n, z
INTEGER strind1, strind2, strind3, strlen, hount, hount2, raninit
INTEGER chopnum, splinterrn, ios
INTEGER traceprof, IMFprof, modelprof, ilim, nrep, jtype
REAL*8 k, PI, ypl, ypn, eps, hl, hadv, dxsav, dxsav2, p0, isodf, traceinpl
REAL*8 xp(KMAXX), xp2(KMAXX), yp(NMAX, KMAXX), yp2(NMAX, KMAXX), voltf
REAL*8 lgphi(intnum), lgrho(intnum), lgr(intnum), lgdphidr(intnum)
REAL*8 rmin, rmax, ystart(NMAX), beta(tracedatnum), rref, a, b, rs
REAL*8 spd2dphi2(intnum), lgvdr2(tracedatnum), lgvdt2(tracedatnum)
REAL*8 trancelgr(tracedatnum), newy(voltN), vdr2pt, vdt2pt
REAL*8 L0, phistop, xi, xf, newx(voltN), lgvd2(intnum), traceoupl
REAL*8 intphipt, hlog(voltN), Evec(voltN), lgtrapg(voltN), tracetrans
REAL*8 dummy(5*voltN), rEfind, rLfind, grmin, grmax, Evec2(voltN+1)
REAL*8 lgLc(intnum), lgEc(intnum), spd2LcdEc2(intnum), hfunc, lgrE
REAL*8 lgrc(intnum), lgcv2(intnum), Emax, Emin, DED(DEDN), gofE
REAL*8 LcplusL0, spdld2gdE2(voltN), rf, Npt, rofphifn, phiofrfn, intgtpt
REAL*8 lgM(intnum), ran2, Mpt, x, lgrhofn, rtbis, lgtrapg2(voltN+1)
REAL*8 objjrad(objnum), objphi(objnum), objE(objnum), objh(objnum)
REAL*8 MChmax, intEpt, MCh, hmax, objhmax(objnum), spd2rcdEc2(intnum)
REAL*8 objrp(objnum), objra(objnum), objrcofE(objnum), j, bisacc
REAL*8 objrcofL(objnum), Mmax, spd2phidr2(intnum), objjellip(objnum)
REAL*8 Mmin, IMFparam1, IMFparam2, IMFmass(IMFnum), sumIMF(IMFnum)
REAL*8 spd2MsumM2(IMFnum), objM, objmass(objnum), IMF, IMFvec(IMFnum)
REAL*8 Nltr(tracedatnum), spd2rdnutr2(tracedatnum), rinit, rmaxpt
REAL*8 d2phidr2(intnum), isodfvec(voltN), newy2(voltN)
REAL*8 xpt, ypt, DEDLcplusL0(DEDN), Emintrace, objvphi(objnum)
REAL*8 ja, jb, spd2sumDEDE2(DEDN)
REAL*8 objvtheta(objnum), objvr(objnum), spd2DEDE2(DEDN), hpt
REAL*8 time1, time2, dphidrof, lgd2phidr2(intnum), lggrvec(voltN)
REAL*8 rmaxtrace, rmintrace, Einfimum, Esupremum
REAL*8 plusminus, gamma, Ept, MCL, trancelgrho(tracedatnum)
REAL*8 dphidr(intnum), newxf, rpt, rhopt, newEsup, DEDEvec(DEDN)
REAL*8 spd2rcdLc2(intnum), spd2vc2dr2(intnum), splgd3phidr3(intnum)
REAL*8 trancelgphi(tracedatnum), spd2rhodr2(tracedatnum), upplim
REAL*8 spd2vdr2dr2(tracedatnum), spd2vdt2dr2(tracedatnum)
REAL*8 new1(tracedatnum), new2(tracedatnum), new3(tracedatnum)
REAL*8 new4(tracedatnum), spd2nutrdr2(tracedatnum)
REAL*8 sumDED(DEDN)
REAL*8 spd2hdsumh2(KMAXX), sumh(KMAXX)
REAL*8 hohmax(KMAXX), inthpt
REAL*8 sumg(KMAXX), gintE(KMAXX), spd2Edsumg2(KMAXX)
CHARACTER*30 savfilename, objfilename, tracename, splintcall
CHARACTER*30 voltname, DEDname
CHARACTER*15 strnum
CHARACTER*650 line
LOGICAL edgereach, validmodel, switch, linin, nonint, valid, now

COMMON /tracepath/ trancelgrho, trancelgr, spd2rhodr2

```

```

COMMON /splintpath/ splintcall
COMMON /errpath/ splinterrn
COMMON /tracerpath/ traceinpl, traceoupl, tracetrans, traceprof
COMMON /refpath/ rref, rf
COMMON /maxpath/ phistop
COMMON /vdpath/ lgtrapg, spdlld2gdE2, Evec
COMMON /gintpath/ intphipt
COMMON /phiandrp/ lgr, lgphi, spd2rdphi2, spd2phidr2, lgdphidr,
* splgd3phidr3
COMMON /minmaxpath/ rmintrace, rmaxtrace, hpt
COMMON /DEDpath/ Ept, spd2DEDdE2, DED, DEDEvec
COMMON /Lcpath/ spd2LcdEc2, lgLc, lgEc
COMMON /L0path/ L0
COMMON /Ecpath/ lgrc, spd2rcdEc2, spd2rcdLc2
COMMON /rhopath/ k, p0, edgereach
COMMON /path/ kmax, kount, dxsav, xp, yp, hount
COMMON /path2/ kmax2, kount2, dxsav2, xp2, yp2, hount2
COMMON /modelpath/ modelprof, IMFprof
COMMON /ELpath/ intEpt, MCL
COMMON /vc2path/ lgc2, spd2vc2dr2
COMMON /IMFpath/ IMFparam1, IMFparam2
COMMON /linpath/ linin
COMMON /nonintpath/ nrep, nonint, upplim
COMMON /hfuncpath/ lgrE, rmaxpt
COMMON /jpath/ ja, jb, jtype
COMMON /ballspath/ now
EXTERNAL poisson, dfvdr2, dfvdt2, rkqs, rkqs2, voltf, H, hfunc, IMFint
EXTERNAL traceint, DEDint, sumDEDint, dfrho, jeans, hint, Eint
validmodel=.false.
switch = .false.
linin = .true.
splinterrn = 0

1  WRITE(6,1001) 'input_model_type:_(polytrope,_king,_dehnen,_or_'
*   'powerlaw)'
  WRITE(6,1000) '1:_polytrope'
  WRITE(6,1000) '2:_king'
  WRITE(6,1000) '3:_dehnen'
  WRITE(6,1000) '4:_powerlaw'
  read(5,*) modelprof

!   bisection variables
bisacc = 1.d-10 ! brackets must be within this absolute tolerance
!               ! of each other before function value at either bracket
!               ! is accepted as root

!   common variables
PI = 3.141592654d0
rinit = 1.d-4 ! minimum radius model properties are tabulated for
rf = 1.d10 ! maximum radius model properties are tabulated for
p0 = 9.d0 ! value of product of constants from dimensionless Poissons
! equation
phistop = -1.d10 ! value of potential at outer edge of system,
!               ! initialised to be negative

!   spline variables
yp1 = 1.d30
ypn = 1.d30

!   model variables
if (modelprof.eq.1) then ! polytrope initialisation (k=2 gives isothermal
! sphere, k=2.5 gives plummer sphere)
  validmodel=.true.
2  WRITE(6,1000) 'input_k,_1<k<=2.5'
  read(5,*) k
  if (k.le.1.d0.or.k.gt.2.5d0) then
    print *, 'invalid_k_selection'
    goto 2
  end if
  rref = 0.d0
  if(k.gt.2.d0) phistop = (2.d0*(k-1.d0)/(k-2.d0)) ! phi only has a maximum for k>2.
  savfilename = 'polytrope_model_k_'
  objfilename = 'polytrope_objct_k_'
  tracename = 'polytrope_trace_k_'
  volname = 'polytrope_distg_k_'
  DEDname = 'polytrope_difED_k_'
  do i = 1,intnum
    lgr(i) = log10(rinit)+(i-1)*log10(rf/rinit)/(1.d0*(intnum-1))
  end do
end if

if (modelprof.eq.2) then ! king 66 stuff (W0-> infinity gives isothermal sphere)
  validmodel=.true.

```

```

3      WRITE(6,1000) 'input_W0,_W0->_0'
      read(5,*) k
      if (k.le.0.d0) then
          print *, 'invalid_W0_selection'
          goto 3
      end if
      rref = 0.d0
      phistop = k
      savfilename = 'king_model_W0_'
      objfilename = 'king_objct_W0_'
      tracename = 'king_trace_W0_'
      voltname = 'king_distg_W0_'
      DEDname = 'king_difED_W0_'
      do i = 1,intnum
          lgr(i) = log10(rinit)+(i-1)*log10(rf/rinit)/(1.d0*(intnum-1))
      end do
end if

if (modelprof.eq.3) then ! dehnen initialisation (gamma=1 gives hernquist,
                        ! gamma=2 gives jaffe)
    validmodel=.true.
4      WRITE(6,1000) 'input_gamma,_0<=_gamma<_3'
      read(5,*) k
      if (k.lt.0.d0.or.k.ge.3.d0) then
          print *, 'invalid_gamma_selection'
          goto 4
      end if
      rref = rinit/1.d1 ! value of radius at which phi = 0 (must be less than rinit)
      if (k.eq.2.d0) then
          phistop = 4.d0*p0*log((rref+1.d0)/rref)
      else if (k.ne.2.d0) then
          phistop = ((2.d0**((4.d0-k))*p0*(1.d0-(rref/
*          (rref+1.d0))**((2.d0-k)))/((3.d0-k)*(2.d0-k)))
      end if
      savfilename = 'dehnen_model_G_'
      objfilename = 'dehnen_objct_G_'
      tracename = 'dehnen_trace_G_'
      voltname = 'dehnen_distg_G_'
      DEDname = 'dehnen_difED_G_'
      do i = 1,intnum
          lgr(i) = log10(rinit)+(i-1)*log10(rf/rinit)/(1.d0*(intnum-1))
      end do
end if

if (modelprof.eq.4) then ! power law stuff (k=2 gives singular isothermal sphere)
    validmodel=.true.
5      WRITE(6,1000) 'input_k,_1<=k<_3'
      read(5,*) k
      if (k.le.1.d0.or.k.ge.3.d0) then
          print *, 'invalid_k_selection'
          goto 5
      end if
      rref = rinit/1.d1 ! value of radius at which phi = 0 (must be less than rinit)
      if (k.gt.2.d0) then
          phistop = (p0*(rref**((2.d0-k)))/((3.d0-k)*(k-2.d0)))
      end if
      savfilename = 'powerlaw_model_k_'
      objfilename = 'powerlaw_objct_k_'
      tracename = 'powerlaw_trace_k_'
      voltname = 'powerlaw_distg_k_'
      DEDname = 'powerlaw_difED_k_'
      do i = 1,intnum
          lgr(i) = log10(rinit)+(i-1)*log10(rf/rinit)/(1.d0*(intnum-1))
      end do
end if

if (validmodel.eqv..false.) then
    WRITE(6,1000) 'invalid_model_type'
    goto 1
end if

18     WRITE(6,1001) 'enter_negative_integer_for_random_number',
*         'generator_seed'
      read(5,*) raninit
      if (raninit.ge.0) goto 18

6      WRITE(6,1000) 'select_tracer_number_density_profile:'
      WRITE(6,1000) '1:_self-consistent'
      WRITE(6,1000) '2:_3_parameter_double_power-law'
      read(5,*) traceprof
      WRITE(6,1000) 'input_maximum_tracer_instantaneous_radius'
      read(5,*) rmaxtrace
      WRITE(6,1000) 'input_minimum_tracer_instantaneous_radius'

```

```

read(5,*) rmintrace
if (traceprof.eq.2) then
  WRITE(6,1000) 'enter_inner_power-law_slope_(gamma)'
  read(5,*) traceinpl
  WRITE(6,1000) 'enter_outer_power-law_slope_(beta)'
  read(5,*) traceoupl
  WRITE(6,1000) 'enter_transition_region_width_parameter_(alpha)'
  read(5,*) tracetran
end if
if (traceprof.ne.1.and.traceprof.ne.2) then
  print *, 'invalid_tracer_number_density_profile_selection'
  goto 6
end if

7  WRITE(6,1000) 'select_tracer_IMF:'
  WRITE(6,1000) '1:_Schechter'
  WRITE(6,1000) '2:_Gaussian'
  WRITE(6,1000) '3:_Powerlaw'
  read(5,*) IMFprof
  if (IMFprof.eq.1) then
    WRITE(6,1000) 'input_Schechter_exponential_cutoff_mass'
    read(5,*) IMFparam1
    WRITE(6,1000) 'input_Schechter_powerlaw_component_slope'
    read(5,*) IMFparam2
  else if (IMFprof.eq.2) then
    WRITE(6,1000) 'input_Gaussian_centre_logarithmic_mass'
    read(5,*) IMFparam1
    WRITE(6,1000) 'input_Gaussian_dispersion'
    read(5,*) IMFparam2
  else if (IMFprof.eq.3) then
    IMFprof = 1
    IMFparam1 = 1.d300
    WRITE(6,1000) 'input_powerlaw_slope'
    read(5,*) IMFparam2
  else
    print *, 'invalid_IMF_selection'
    goto 7
  end if
8  WRITE(6,1000) 'input_logarithmic_maximum_initial_tracer_mass'
  read(5,*) Mmax
  WRITE(6,1000) 'input_logarithmic_minimum_initial_tracer_mass'
  read(5,*) Mmin
  if (Mmax.le.Mmin) then
    print *, 'maximum_mass_must_be_greater_than_minimum_mass'
    goto 8
  end if

11 WRITE(6,1000) 'select_circularity_function'
  WRITE(6,1000) '1:_j(h)=-h*{-2a}_(beta=a)'
  WRITE(6,1000) '2:_j(h)=-exp(-ah)'
  WRITE(6,1000) '3:_j(h)=-exp(-0.5*((h-a)/b)^2)'
  WRITE(6,1000) '4:_j(h)=-1-(h^a)[1-ln(h^a)]'
  WRITE(6,1000) '5:_j(h)=-0.5[1+h*sin(ah)]'
  read(5,*) jtype
  valid = .false.
  do i = 1,5
    if (jtype.eq.i) valid = .true.
  end do
  if (valid.eqv..false.) then
    WRITE(6,1000) 'invalid_choice.'
    goto 11
  end if
  if (jtype.eq.1) then
20  WRITE(6,1000) 'enter_a_(-infty<a<1)'
    read(5,*) ja
    if (ja.ge.1) goto 20
  end if
  if (jtype.eq.2) then
21  WRITE(6,1000) 'enter_a_(a>0)'
    read(5,*) ja
    if (ja.le.0) goto 21
  end if
  if (jtype.eq.3) then
22  WRITE(6,1000) 'enter_a_and_b_(0<=a<=1,b>0)'
    read(5,*) ja,jb
    if (ja.lt.0.d0.or.ja.gt.1.d0.or.jb.le.0.d0) goto 22
  end if
  if (jtype.eq.4) then
23  WRITE(6,1000) 'enter_a_(a>0)'
    read(5,*) ja
    if (ja.le.0.d0) goto 23
  end if
  if (jtype.eq.5) then

```

```

        WRITE(6,1000) 'enter _a_(-infy<a<infy)'
        read(5,*) ja
    end if

24  WRITE(6,1000) 'enter _L0_(L0>=0)'
    read(5,*) L0
    if (L0.lt.0.d0) goto 24

    WRITE (strnum,*) (k+5.d0*(1.d1**(-1.d0-1.d0*(dcmlpnts))))

    strind1 = index(strnum, '.')
    strnum = strnum(1:strind1+dcmlpnts)

    strlen = len(strnum) ! file naming stuff
    strind1 = index(savfilename, '_') ! find position of first blank
    ! space in savfilename
    strind2 = index(strnum, '.') ! find position of decimal place in
    ! W0 string
    9  strind3 = index(strnum, '_') ! find position of first blank space
    ! in W0 string
    if (strind3.eq.1) then ! remove all leading blank spaces
        strnum=strnum(strind3+1:strlen)
        goto 9
    end if ! loops until first character of strnum is not a blank space
    strind2 = index(strnum, '.') ! find new position of decimal palce
    strind3 = index(strnum, '_') ! find new position of first blank space, now
    ! somewhere at end of string
    if (strind3.eq.strind2+1) then ! if there is no zeros after the decimal
    ! place, add as many as are required
        do i = 1,dcmlpnts
            strnum = strnum(1:(strind2+i-1)) // '0'
        end do
        strind3 = index(strnum, '_') ! find position of first blank
        ! space again
    end if
    ! objfilemae, tracername, and voltname deliberately have same
    ! initial character length as savfilename
    savfilename=savfilename(1:strind1-1)//strnum(1:strind3-1)//'.out'
    objfilename=objfilename(1:strind1-1)//strnum(1:strind3-1)//'.out'
    tracername=tracername(1:strind1-1)//strnum(1:strind3-1)//'.out'
    voltname = voltname(1:strind1-1)//strnum(1:strind3-1)//'.out'
    DEDname = DEDname(1:strind1-1)//strnum(1:strind3-1)//'.out'

    CALL cpu_time(timel)
    print *, 'Generating_model...'

    OPEN(4,FILE=savfilename,STATUS='OLD',IOSTAT=ios,ERR=746)
    i = 0
    do n=1,2147483646
        read(4,1000,end=747) line
        if (line(1:1).ne. '#') then
            i = i + 1
            read(line,*) lgphi(i),lgr(i),lgdphidr(i),
*                lgrho(i),lgd2phidr2(i),lgcv2(i),
*                lgM(i),lgvd2(i),lgEc(i),lgLc(i)
        end if
    end do
747 CLOSE(4)
    rf = 1.d1**lgr(intnum)
    do i = 1,intnum
        ! linear first derivative
        dphidr(i) = (lgphi(i)-lgr(i))+log10(lgdphidr(i))
        ! linear second derivative
        d2phidr2(i) = p0*(1.d1**lgrho(i))-2.d0*(1.d1**
*            (lgphi(i)-2.d0*lgr(i)))*lgdphidr(i)
        ! dimensionless system mass
        lgM(i) = lgphi(i)+lgr(i)+log10(lgdphidr(i))
        ! consider circular orbits at the same radii as iterated radii throughout system
        lgrc(i) = lgr(i)
        ! squared circular speed
        lgcv2(i) = lgphi(i)+log10(lgdphidr(i))
        ! energy of circular orbit
        lgEc(i) = lgphi(i)+log10(1.d0+5.d-1*lgdphidr(i))
        ! angular momentum of circular orbit
        lgLc(i) = lgrc(i)+5.d-1*lgphi(i)+5.d-1*log10(lgdphidr(i))
    end do
    CALL spline(lgEc,lgrc,intnum,yp1,ypn,spd2rcdEc2)
    CALL spline(lgLc,lgrc,intnum,yp1,ypn,spd2rcdLc2)
    CALL spline(lgEc,lgLc,intnum,yp1,ypn,spd2LcdEc2)
    CALL spline(lgphi,lgr,intnum,yp1,ypn,spd2rdphi2)
    CALL spline(lgr,lgphi,intnum,yp1,ypn,spd2phidr2)
    CALL spline(lgr,lgdphidr,intnum,yp1,ypn,splgd3phidr3)
    CALL spline(lgr,lgcv2,intnum,yp1,ypn,spd2vc2dr2)

```

```

746  if (ios.ne.0) then
10    if (modelprof.eq.1.or.modelprof.eq.2) then
        nvar = 2
        kmax = 0
        eps = 1.d-8
        hl = 1.d-10
        hadv = 1.d-10 ! this is how much the integration variable is
                        ! pushed along by if the integration gets stuck
        xi = -5.d1
        do i = 1,intnum
            ystart(1) = 2.d0*xi+log10(p0/6.d0) ! initial value of potential
            ystart(2) = 2.d0 ! initial value of first deivative of potential
            xf = lgr(i)
            CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,poisson
*             ,rkqs)
            if (modelprof.eq.2.and.ystart(1).gt.log10(phistop)) then
                rmin = lgr(i-1)
                rmax = lgr(i)
                nvar = 2
                kmax = 0 ! = KMAXX
                eps=1.d-7
                hl=1.d-10
                hadv=1.d-10
                do z = 1,itnum ! iterate over last 2 steps, to find radius of boundary
                    xi = -5.d1
                    xf = (rmin + (rmax-rmin)*(z-1)/(1.d0*(itnum-1)))
                    ystart(1) = 2.d0*xi+log10(p0/6.d0)
                    ystart(2) = 2.d0
                    CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,
*                     poisson,rkqs)
                    if (ystart(1).gt.log10(phistop)) then
                        ! if edge is reached, replace rf with previous radius as tidal boundary
                        rf = 1.d1**((rmin + (rmax-rmin)*(z-2)/(1.d0*(itnum-1)))
                        do n = 1,intnum ! define new radii vector using new rf
                            lgr(n)=log10(rinit)+(n-1)*log10(rf/rinit)/
*                             (1.d0*(intnum-1))
                        end do
                        goto 10 ! then go back and calculate phi for
                                ! intnum radii between rinit and new rf
                    end if
                end do
                end if
                lgdphidr(i) = ystart(2)
                lgphi(i) = ystart(1)
                lgrho(i) = lgrhofn(ystart(1),xf)
            end do
            else if (modelprof.eq.3.or.modelprof.eq.4) then
                do i = 1,intnum
                    lgphi(i) = phiofrfn(lgr(i))
                    lgdphidr(i) = dphidrofr(lgr(i))
                    lgrho(i) = lgrhofn(lgphi(i),lgr(i))
                end do
            end if

            do i = 1,intnum
                dphidr(i) = lgphi(i)-lgr(i)+log10(lgdphidr(i)) ! linear first derivative
                d2phidr2(i) = p0*(1.d1**lgrho(i))-2.d0*(1.d1**
*                 (lgphi(i)-2.d0*lgr(i))*lgdphidr(i) ! linear second derivative
                lgd2phidr2(i) = p0*1.d1**((2.d0*lgr(i)+lgrho(i)-lgphi(i))
                lgd2phidr2(i) =(lgd2phidr2(i)-lgdphidr(i))*(lgdphidr(i)+1.d0))
*                 *log(1.d1) ! logarithmic second derivative
            end do

            do i = 1,intnum
                ! system mass
                lgM(i) = lgphi(i)+lgr(i)+log10(lgdphidr(i))
                ! radii of system circular orbits
                lgrc(i) = lgr(i)
                ! circular speed
                lgcv2(i) = lgphi(i)+log10(lgdphidr(i))
                ! energy of circular orbit
                lgEc(i) = lgphi(i)+log10(1.d0+5.d-1*lgdphidr(i))
                ! angular momentum of circular orbit
                lgLc(i) = lgrc(i)+5.d-1*lgphi(i)+5.d-1*log10(lgdphidr(i))
            end do

            CALL spline(lgEc,lgrc,intnum,ypl,ypn,spd2redEc2)
            CALL spline(lgLc,lgrc,intnum,ypl,ypn,spd2redLc2)
            CALL spline(lgEc,lgLc,intnum,ypl,ypn,spd2LcdEc2)
            CALL spline(lgphi,lgr,intnum,ypl,ypn,spd2rdphi2)
            CALL spline(lgr,lgphi,intnum,ypl,ypn,spd2phidr2)
            CALL spline(lgr,lgdphidr,intnum,ypl,ypn,splgd3phidr3)

```

```

CALL spline(lgr,lgcv2,intnum,yp1,ypn,spd2vc2dr2)

nvar = 1
kmax = 0
eps = 1.d-7
h1 = 1.d-10
hadv = 1.d-10 ! this is how much the integration variable is pushed
                ! along by if the integration gets stuck
do i = 1,intnum ! integrals for host galaxy isotropic velocity dispersion
  xi = lgr(i)
  xf = 4.d1! in log space, this should be large enough as a stand in for infinity
  if (modelprof.eq.2) xf = log10(rf)
  if (modelprof.eq.2.and.i.eq.intnum) then
    lgvd2(i) = -1.d40
  else
    ystart(1) = 0.d0
    CALL odeint(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,jeans,
*                                     rkqs)
    lgvd2(i) = log10(log(1.d1)*ystart(1))-lgrho(i)
  end if
end do

OPEN(7,file=savfilename)
if(modelprof.eq.1.or.modelprof.eq.2) then
  WRITE(7,1006) '#phi(0)=',0.d0,',_rref=',rref
else if (modelprof.eq.3.or.modelprof.eq.4) then
  WRITE(7,1006) '#phi(rref)=',0.d0,',_rref=',rref
end if
WRITE(7,1007) '#column_1:_dimensionless_potential=,log[(Phi-',
* 'Phi(rinit))/sigma_0^2]'
WRITE(7,1000) '#column_2:_dimensionless_radius=log[r/r_0]'
WRITE(7,1001) '#column_3:_dimensionless_first_derivative_of',
* '_potential_w.r.t._radius=dlog[dphi/sigma_0^2]/dlog[r/r_0]'
WRITE(7,1001) '#column_4:_dimensionless_density=',
* '_log[rho/rho(0)]'
WRITE(7,1001) '#column_5:_dimensionless_second_derivative_of',
* '_potential_w.r.t._radius=d2log[phi/sigma_0^2]/dlog[r/r_0]^2'
WRITE(7,1001) '#column_6:_dimensionless_local_circular_speed_',
* 'squared=log[(r/sigma_0^2)*dphi/dr]'
WRITE(7,1001) '#column_7:_dimensionless_mass_interior_to_r_',
* '=log[GM(<r)/(sigma_0^2*r_0)]'
WRITE(7,1001) '#column_8:_dimensionless_squared_isotropic_',
* 'velocity_dispersion=log[sigma^2/sigma_0^2]'
WRITE(7,1001) '#column_9:_dimensionless_energy_of_circular_',
* 'orbit_with_radius_r=log[Ec/sigma_0^2]'
WRITE(7,1001) '#column_10:_dimensionless_angular_momentum_of_',
* '_circular_orbit_with_radius_r=log[Lc/(r_0*sigma_0)]'
do i = 1,intnum
  WRITE(7,1002) lgphi(i),lgr(i),lgdphidr(i),
*               lgrho(i),lgd2phidr2(i),lgcv2(i),
*               lgM(i),lgvd2(i),lgEc(i),lgLc(i)
end do
CLOSE(7)
end if

CALL cpu_time(time2)
print *, 'Model_generation_done,_data_saved_to_',savfilename
print *, 'Time_taken=',time2-time1
print *, 'Solving_integral_equation...'

OPEN(4,FILE=voltname,STATUS="OLD",IOSTAT=ios,ERR=745)
i = 0
do n=1,2147483646
  read(4,1000,end=748) line
  if (line(1:1).ne.'#') then
    i = i + 1
    read(line,*) Evec(i),lggrvec(i),isodfvec(i),lgtrapg(i)
  end if
end do
CLOSE(4)
! make definitions not otherwise made when reading in rather than reproducing results
Emax = phiofrfn(log10(rf))
Emin = lgEc(2)
Einimum = Evec(1)
Esupremum = Evec(voltN)
grmin=rofpfhn(Einimum)
grmax=rofpfhn(Esupremum)
if (rmintrace.lt.1.d1**grmin) rmintrace = 1.d1**grmin
if (rmaxtrace.gt.1.d1**grmax) rmaxtrace = 1.d1**grmax

do i = 1,tracedatnum
  tracelgr(i)=log10(rmintrace)+(i-1)*log10(rmaxtrace/rmintrace)/

```

```

*          (1.d0*(tracedatnum-1))
  trancelgphi(i) = phiofrfn(trancelgr(i))
end do
! if trancelgphi(tracedatnum) is greater than Esupremum,
! trancelgphi(tracedatnum) = Esupremum
trancelgphi(tracedatnum) = min(trancelgphi(tracedatnum),Esupremum)
! if trancelgphi(1) is smaller than Einfimum, trancelgphi(1) = Einfimum
trancelgphi(1) = max(trancelgphi(1),Einfimum)

745 if (ios.ne.0) then
  CALL cpu_time(timel)
  Emax = phiofrfn(log10(rf))
  Emin = lgEc(2) ! need to have energies used in volterra equation bounded
                  ! by lgEc(1) and Emax so that Lc(E) can be interpolated.
                  ! Set to lgEc(2) so that Evec is still bounded after
                  ! monotonic call
  rs = 1.d1**rofphifn(Emin)
  do i = 1,voltN+1 ! potentials are defined as phi of linearly spaced r's
    Evec2(i) = phiofrfn(log10(rs)+(i-1)*log10(rf/rs)/
*          (1.d0*(voltN)))
  end do
  ! ensure the vector is monotonic to within machine accuracy
  CALL monotonic(Evec2,voltN+1,dummy,1.d-13)
  do i = 1,voltN+1
    Evec2(i) = dummy(i)
  end do
  do i = 1,voltN
    hlog(i) = Evec2(i+1)-Evec2(i)
  end do
  CALL voltrap(voltN+1,Evec2,hlog,lgtrapg2,voltf,H)
  do i = 1,voltN
    ! because of singular Kernel, trapezoidal rule is unable to calculate g at Evec(1),
    ! so discard it and shift all associated vectors down by 1
    Evec(i) = Evec2(i+1)
    lgtrapg(i) = lgtrapg2(i+1)
  end do
  CALL chop(Evec,lgtrapg,voltN,a,b,newx,newy,chopnum)

  Einfimum = a
  Esupremum = b
  grmin=rofphifn(Einfimum)
  grmax=rofphifn(Esupremum)
  if (grmax.gt.log10(rf)) grmax = log10(rf)

  do i = 1,voltN
    write (88,*) Evec(i),rofphifn(Evec(i)),lgtrapg(i)
  end do
  do i = 1,chopnum
    write (89,*) newx(i),rofphifn(newx(i)),newy(i)
  end do

  CALL spline(newx,newy,chopnum,yp1,ypn,newy2)
  do i=1,voltN
    xpt=phiofrfn(grmin+(i-1)*(grmax-grmin)/(voltN-1))
    splintcall = 'chop'
    CALL splintcheck(newx,newy,newy2,chopnum,xpt,ypt,.true.)
    Evec(i) = xpt
    lgtrapg(i) = ypt
  end do
  ! ensure the vector is monotonic to within machine accuracy
  CALL monotonic(Evec,voltN,dummy,1.d-13)
  do i = 1,voltN
    Evec(i) = dummy(i)
  end do

  print *, 'log[Emin]=' ,Emin, '_log[Emax]=' ,Emax
  print *, 'Einfimum=' ,Einfimum, '_Esupremum=' ,Esupremum
  print *, 'log[grmin]=' ,grmin, '_log[grmax]=' ,grmax

  do i = 1,voltN
    lgtrapg(i) = log10(lgtrapg(i)) ! log[g(E)]
    lggrvec(i) = rofphifn(Evec(i)) ! r(Phi=E)
  end do
  CALL spline(Evec,lgtrapg,voltN,yp1,ypn,spdl2gdE2)

  if (rmintrace.lt.1.d1**grmin) rmintrace = 1.d1**grmin
  if (rmaxtrace.gt.1.d1**grmax) rmaxtrace = 1.d1**grmax

  do i = 1,tracedatnum
    trancelgr(i)=log10(rmintrace)+(i-1)*log10(rmaxtrace/rmintrace)/
*          (1.d0*(tracedatnum-1))
    trancelgphi(i) = phiofrfn(trancelgr(i))

```



```

end do
! account for possibility of roundoff causing problems when tracer
! population is spread over all numerically resolved radii
tracelgphi(tracedatnum) = min(tracelgphi(tracedatnum), Esupremum)
tracelgphi(1) = max(tracelgphi(1), Einfimum)

OPEN(7, file=voltname)
WRITE(7,1001) '#column_1:_dimensionless_energy_=_',
*      'log[E/sigma_0^2]',
WRITE(7,1001) '#column_2:_dimensionless_radius_corresponding_',
*      'to_potential_of_energy_E_=_log[r(phi=E)/r_0]',
WRITE(7,1001) '#column_3:_dimensionless_isotropic_f(E)_=_',
*      'log[f(E)/(r_0*sigma_0)^3]',
WRITE(7,1001) '#column_4:_dimensionless_g(E)_=_',
*      'log[g(E)/(r_0*sigma_0)^3]',
do i = 1, voltN
  WRITE(7,1010) Evec(i), lggrvec(i), lgtrapg(i), lgtrapg(i)
end do
CLOSE(7)
end if

CALL cpu_time(time2)
print *, 'distribution_function_done, _output_saved_',
*'to_', voltname
print *, 'Time_taken_=_', time2-time1

print *, 'Calculating_tracer_properties...'
CALL cpu_time(time1)

OPEN(4, FILE=tracename, STATUS='OLD', IOSTAT=ios, ERR=744)
i = 0
do n=1, 2147483646
  read(4,1000, end=749) line
  if (line(1:1).ne. '#') then
    i = i + 1
    read(line, *) tracelgphi(i), tracelgrho(i), tracelgr(i),
*      lgvdr2(i), lgvdt2(i), beta(i), Nltr(i)
  end if
end do
749 CLOSE(4)

744 if (ios.ne.0) then
! actual tracer density profile as calculated from the distribution function
eps = 1.d-5
nvar = 1
kmax = 0 ! = KMAXX
switch = .false.
nonint = .false.
ilim = tracedatnum

do i = 1, tracedatnum
  nrep = 0
  xi = tracelgphi(i)
  xf = Esupremum ! greatest energy available to tracers
  h1 = (xf-xi)/1.d8 ! set stepsize to be some fraction
! of the current integration range

  hadv = h1
  intphipt = tracelgphi(i)
  upplim = xf
  ystart(1) = 0.d0
  if (switch.eqv..false.) then
    CALL odeint(ystart, nvar, xi, xf, eps, h1, hadv, nok, nbad, dfrho,
*      rkqs)
155 * if (ystart(1).eq.0.d0.or.nonint.eqv..true.) then ! as number density
! monotonically decreases, it should only
! hit 0 at the last radius, where the tracer
! system ceases to be defined
tracelgrho(i) = logzero ! dont want to save NANs
lgvdt2(i) = logzero ! as velocity dispersions depend on the
! reciprocal of density, NANs also a problem here
lgvdr2(i) = logzero
if (switch.eqv..false.) then ! first time the numer density drops to zero
newxf = tracelgr(i-1) ! save the previous radius
newEsup = tracelgphi(i-1) ! save the previous value of the potential
ilim = i-1 ! save the previous iteration number
print *, 'numerical_roundoff_rendering_integrals_',
*      'unsolvable._Reducing_outermost_radius_to:',
*      1.d1**newxf
end if
switch = .true. ! flip the switch, so that no more integrals are attempted
else ! otherwise calculate density and velocity dispersions normally

```

```

tracelgrrho(i)=log10(4.d0*PI*log(1.d1)*ystart(1))-
      tracelgr(i)
*      ! radial velocity dispersion integral from distribution function
      ystart(1)=0.d0
      CALL odeint(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,dfvdr2
*      ,rkqs)
      if (ystart(1).eq.0.d0) then
        nonint = .true.
        goto 155
      else
        lgvdr2(i) = log10(4.d0*PI*log(1.d1))+log10(ystart(1))-
*      tracelgrrho(i)-3.d0*tracelgr(i)
      end if
      ! tangential velocity dispersion integral from distribution function
      ystart(1)=0.d0
      CALL odeint(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,dfvdt2
*      ,rkqs)
      if (ystart(1).eq.0.d0) then
        nonint = .true.
        goto 155
      else
        lgvdt2(i) = log10(4.d0*PI*log(1.d1))+log10(ystart(1))-
*      tracelgrrho(i)-3.d0*tracelgr(i)
      end if
      beta(i) = 1.d0-5.d-1*1.d1**((lgvdt2(i)-lgvdr2(i))
      end if
    else
      tracelgrrho(i) = logzero
      lgvdt2(i) = logzero
      lgvdr2(i) = logzero
      beta(i) = 1.d3 ! set beta to something unphysical
    end if
  end do

  if (switch.and.ilim.ne.tracedatnum-1) then ! if density has fallen to zero,
                                              ! define new xf
    if (rmaxtrace.gt.1.d1**newxf) then ! still need rmaxtrace < outer limit
      rmaxtrace = 1.d1**newxf
      ! build interpolation tables so that tracedatnum points can be saved
      ! into the array between rmintrace and new rmaxtrace
      CALL spline(tracelgr,tracelgrrho,ilim,yp1,ypn,spd2rhodr2)
      CALL spline(tracelgr,lgvdr2,ilim,yp1,ypn,spd2vdr2dr2)
      CALL spline(tracelgr,lgvdt2,ilim,yp1,ypn,spd2vdt2dr2)
      do i = 1,tracedatnum ! final element of vector is 0 or logzero
        rpt = log10(rmintrace)+(i-1)*log10(rmaxtrace/rmintrace)
*      /(1.d0*(tracedatnum-1)) ! define tracedatnum new tracer
*      ! radii between rmintrace and
*      ! rmaxtrace
        new1(i) = rpt ! save new radii to a temporary array since old ones
*      ! still in use for interpolations
        splintcall = 'tracelgr_tracelgrrho'
        CALL splintcheck(tracelgr,tracelgrrho,spd2rhodr2,ilim,rpt,
*      ,rhopt,.true.)
        new2(i) = rhopt ! save new densities to a temporary array since old
*      ! ones still in use for interpolations
        tracelgphi(i) = phiofrfn(rpt) ! define new potentials at new radii
        splintcall = 'tracelgr_lgvdr2'
        ! interpolate on radial velocity dispersion as a function of radius
        ! so that there are tracedatnum points
        CALL splintcheck(tracelgr,lgvdr2,spd2vdr2dr2,ilim,rpt,
*      ,vdr2pt,.true.)
        new3(i) = vdr2pt ! save new radial velocity dispersions to a
*      ! temporary array since old ones still in use for
*      ! interpolations
        splintcall = 'tracelgr_lgvdt2'
        ! interpolate on tangential velocity dispersion as a function of
        ! radius so that there are tracedatnum points
        CALL splintcheck(tracelgr,lgvdt2,spd2vdt2dr2,ilim,rpt,
*      ,vdt2pt,.true.)
        new4(i) = vdt2pt ! save new tangential velocity dispersions to a
*      ! temporary array since old ones still in use for
*      ! interpolations
      end do
      tracelgphi(tracedatnum) = min(tracelgphi(tracedatnum),
*      newEsup)
    do i = 1,tracedatnum
      tracelgr(i) = new1(i) ! overwrite old radii with new ones defined
*      ! between rmintrace and new rmaxtrace
      tracelgrrho(i) = new2(i) ! overwrite old densities with new ones
*      ! only up to new rmaxtrace
      lgvdr2(i) = new3(i) ! overwrite old velocity dispersions with new
*      ! ones only up to new rmaxtrace
      lgvdt2(i) = new4(i)
    end do
  end if

```

```

        ! define new anisotropy parameter beta using new velocity
        ! dispersions
        beta(i) = 1.d0-5.d-1*1.d1** (lgvdt2(i)-lgvdr2(i))
    end do
    end if
end if

! build interpolation tables for use in tracerhofn
CALL spline(tracelgr, tracelgrho, tracedatnum, ypl, ypn, spd2rhodr2)

!
    tracer number interior to radius r integrations
    eps=1.d-8
    nvar = 1
    kmax = 0 ! = KMAXX
    do i = 1, tracedatnum
        xi = rmintrace ! by definition, there are no tracer objects inside of rmintrace
        xf = 1.d1**tracelgr(i)
        h1 = (xf-xi)/1.d4
        hadv = h1
        ystart(1)=0.d0
        CALL odeint(ystart, nvar, xi, xf, eps, h1, hadv, nok, nbad, traceint
*           , rkqs)
        Nltr(i) = ystart(1)
    end do

    CALL spline(Nltr, tracelgr, tracedatnum, ypl, ypn, spd2rdntr2)
    CALL spline(tracelgr, Nltr, tracedatnum, ypl, ypn, spd2nutdr2)

    OPEN(7, name=tracename)
    WRITE(7,1001) '#column_1:_dimensionless_potential_=',
*           '_log [(Phi-Phi(0))/sigma_0^2]',
    WRITE(7,1001) '#column_2:_dimensionless_number_density_=',
*           '_log[nu/nu(0)]',
    WRITE(7,1000) '#column_3:_dimensionless_radius_=_log[r/r_0]',
    WRITE(7,1001) '#column_4:_dimensionless_radial_velocity_=',
*           '_dispersion_squared_=_log[sigma_r^2/sigma_0^2]',
    WRITE(7,1001) '#column_5:_dimensionless_tangential_=',
*           '_velocity_dispersion_squared_=_log[sigma_t^2/sigma_0^2]',
    WRITE(7,1001) '#column_6:_tracer_anisotropy_parameter_=',
*           '_1-0.5*sigma_t^2/sigma_r^2',
    WRITE(7,1000) '#column_7:_number_interior_to_r_=_[N(<r/r_0)]',
    do i = 1, tracedatnum
        WRITE(7,1003) tracelgphi(i), tracelgrho(i), tracelgr(i),
*           lgvdr2(i), lgvdt2(i), beta(i), Nltr(i)
    end do
    CLOSE(7)
end if

CALL cpu_time(time2)
print *, 'tracer_properties_done, _data_saved_to_', tracename
print *, 'Time_taken=', time2-time1
print *, 'Calculating_differential_energy_distribution...'
CALL cpu_time(time1)

OPEN(4, FILE=DEDname, STATUS="OLD", IOSTAT=ios, ERR=743)
i = 0
do n=1, 2147483646
    read(4,1000, end=750) line
    if (line(1:1).ne. '#') then
        i = i + 1
        read(line, *) DEDEvec(i), DED(i), sumDED(i), DEDLcplusL0(i)
    end if
end do
750 CLOSE(4)

Emintrace = DEDEvec(1)

743 if (ios.ne.0) then
    ! differential energy distribution integrations
    eps=1.d-7
    nvar = 1
    kmax = 0 ! = KMAXX
    Emintrace = tracelgphi(1)
    xi = rmintrace ! all tracer objects are generated between rmintrace and rmaxtrace
    ! with a minimum energy of Emintrace, and below this DED = 0

    do i = 1, DEDN
        Ept=Emintrace+((i-1)/(1.d0*(DEDN-1)))*(Esupremum-Emintrace)
        DEDEvec(i) = Ept
        xf = min(1.d1**rofphifn(Ept), rmaxtrace)
        h1 = (xf-xi)/1.d4
        hadv = h1
        ystart(1) = 0.d0
    end do

```

```

      CALL odeint(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,DEDint,
*          rkqs)
      DED(i) = ((4.d0*PI)**2.d0)*gofE(DEDEvec(i))*
*          LcplusL0(DEDEvec(i))*log(1.d1)*(1.d1**DEDEvec(i))
      DED(i) = DED(i)*ystart(1)
    end do

    ! integrating for the cumulative differential energy distribution (total
    ! area under curve)
    CALL spline(DEDEvec,DED,DEDN,yp1,ypn,spd2DEDdE2)
    xi = Emintrace ! only normalise curve to area where tracers are generated
    do i = 1,DEDN
      DEDLcplusL0(i) = LcplusL0(DEDEvec(i))
      xf = DEDEvec(i)
      h1 = (xf-xi)/1.d4
      hadv = h1
      ystart(1) = 0.d0
      CALL odeint(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,sumDEDint,
*          rkqs)
      sumDED(i) = ystart(1)
    end do
    CALL monotonic(sumDED,DEDN,dummy,1.d-15)
    do i = 1,DEDN
      sumDED(i) = dummy(i)
    end do
    CALL spline(DEDEvec,sumDED,DEDN,yp1,ypn,spd2sumDEDdE2)

    OPEN(7,file=DEDname)
    WRITE(7,1001) '#column_1:_dimensionless_energy_=_',
*      'log[E/sigma_0^2]'
    WRITE(7,1001) '#column_2:_dimensionless_differential_energy_',
*      '_distribution_'
    WRITE(7,1001) '#column_3:_dimensionless_cumulative_',
*      'differential_energy_distribution_'
    WRITE(7,1001) '#column_4:_dimensionless_circular_angular_',
*      'momentum,_Lc(E)+L0_'
    do i = 1,DEDN
      WRITE(7,1010) DEDEvec(i),DED(i),sumDED(i),
*          DEDLcplusL0(i)
    end do
    CLOSE(7)
  end if

  CALL cpu_time(time2)
  print *, 'diff._en._dist._done,_data_saved_to_',DEDname
  print *, 'Time_taken_=',time2-time1

  ! building initial mass function tables and interpolation arrays
  hadv = 1.d-10
  h1 = 1.d-10
  nvar = 1
  eps = 1.d-5
  do i = 1,IMFnum
    ystart(1) = 0.d0
    xi = Mmin
    xf = Mmin+(Mmax-Mmin)*(i-1)/(1.d0*(IMFnum-1))
    CALL odeint(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,IMFint,
*      rkqs)
    sumIMF(i) = ystart(1) ! cumulative mass distribution
    IMFmass(i) = xf ! mass arrays
    IMFvec(i) = IMF(xf) ! number per unit mass at mass coordinate
  end do
  do i = 1,IMFnum
    ! normalise function to have unit area
    IMFvec(i) = IMFvec(i)/sumIMF(IMFnum)
    ! normalise cumulative distribution to be between 0 and 1
    sumIMF(i) = sumIMF(i)/sumIMF(IMFnum)
  end do
  CALL monotonic(sumIMF,IMFnum,dummy,1.d-15)
  do i = 1,IMFnum
    sumIMF(i) = dummy(i)
  end do
  CALL spline(sumIMF,IMFmass,IMFnum,yp1,ypn,spd2MdsumM2)

  OPEN(8,file='IMF.out')
  do i = 1,IMFnum
    WRITE(8,1004) IMFmass(i),IMFvec(i),sumIMF(i)
  end do
  CLOSE(8)

  print *, 'Beginning_object_generation...'
  OPEN(4,FILE=objfilename,STATUS='OLD',IOSTAT=ios,ERR=742)

```

```

CLOSE(4)
! Monte Carlo sampling
742 if (ios.ne.0) then
    CALL cpu_time(time1)
    do n = 1,KMAXX
        xp(n) = 0.d0
        yp(1,n) = 0.d0
    end do
    do i = 1,objnum
16      Npt = ran2(raninit)*Nltr(tracedatnum) ! sample a radius
        splintcall = 'Nltr_tracelgr'
        CALL splintcheck(Nltr,tracelgr,spd2rdnutr2,tracedatnum,Npt,x,
*          .false.)
25      xi = phiofrfn(x) ! calculate potential at that radius
        xf = Esupremum
        kmax = KMAXX
        nvar = 1
        hadv = 1.d-10
        hl = 1.d-10
        eps = 1.d-7
        intphipt = xi
        dxsav = 0.d0
        ystart(1) = 0.d0
        ! integrate energy probability distribution given a potential
        CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,Eint,rkqs)
        do n = 1,kount
            sumg(n) = yp(1,n)/yp(1,kount)
            gintE(n) = xp(n)
            yp(1,n) = 0.d0
            xp(n) = 0.d0
        end do

        CALL spline(sumg,gintE,kount,yp1,ypn,spd2Edsumg2)

        intgpt = ran2(raninit)*sumg(kount) ! sample an energy
        splintcall = 'sumg_gintE_sampling'
        CALL splintcheck(sumg,gintE,spd2Edsumg2,kount,intgpt,intEpt
*          ,.true.)

        x = 1.d1**x ! x is now linear radius

        MChmax = hmax(intEpt,intphipt)
        objphi(i) = 1.d1**intphipt
        objrad(i) = x
        objhmax(i) = MChmax
        objE(i) = 1.d1**intEpt
15      if (jtype.eq.1) then
            xi = 0.d0
            xf = 1.d0-1.d-10
            kmax = KMAXX
            nvar = 1
            hadv = 1.d-10
            hl = 1.d-10
            eps = 1.d-7
            dxsav = 0.d0
            ystart(1) = 0.d0
            ! integrate h probability distribution
            CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,hint,
*              rkqs)

            do n = 1,kount
                hohmax(n) = xp(n)
                sumh(n) = yp(1,n)/yp(1,kount)
                xp(n) = 0.d0
                yp(1,n) = 0.d0
            end do
            CALL spline(sumh,hohmax,kount,yp1,ypn,spd2hdsumh2)
            inthpt = ran2(raninit)*sumh(kount) ! sample an h
            splintcall = 'sumh_hohmax_hint'
            CALL splintcheck(sumh,hohmax,spd2hdsumh2,kount,inthpt,MCh
*              ,.true.)

            MCh = MChmax*sqrt(MCh)
        else ! otherwise use rejection method
            xf = ran2(raninit)
            MCh = MChmax*sin(xf*PI/2.d0)
            xi = ran2(raninit)
            if (xi*MChmax.gt.MCh*j(MCh)) goto 15
        end if
        objh(i) = MCh
        if (MCh.gt.1.d0) then
            ! sampled r and E may combine to give circular orbit. Numerical roundoff
            ! in hmaxcalculation may cause it to be slightly greater than 1. when
            ! this conspires with a random number very close to 1, h may also be
            ! greater than 1. In this very unlikely event, resample for h
            print *, 'h>1_in_sampling', '_h=',MCh, '_hmax=',MChmax

```

```

        print *, 'resampling...'
        goto 15
    end if
    MCL = MCh*LcplusL0(intEpt)
    objrcofE(i) = rEfind(1.d1**intEpt)
    objrcofL(i) = rLfind(MCL)
    rmaxpt = rofphifn(intEpt)
    lgrE = log10(objrcofE(i))

    if (ran2(raninit).lt .5.d-1) then
        plusminus = -1.d0
    else
        plusminus = 1.d0
    end if
    objvr(i) = plusminus*(LcplusL0(intEpt)/x)*
*      ((MChmax**2.d0)-(MCh**2.d0))**5.d-1
    gamma = ran2(raninit)*2.d0*PI
    objvphi(i) = (LcplusL0(intEpt)*MCh/x)*sin(gamma)
    objvtheta(i) = (LcplusL0(intEpt)*MCh/x)*cos(gamma)

    ! if the pericentre lies inside of rinit, rtbis returns with -1.d40.
    ! When this happens, resample
    objrp(i) = rtbis(hfunc,lgrE,log10(rinit),bisacc)
    if (objrp(i).eq.-1.d40) goto 16
    objra(i) = rtbis(hfunc,lgrE,rmaxpt,bisacc)
    if (objra(i).eq.-1.d40) goto 16

    objellip(i) = (1.d0-1.d1**((objrp(i)-objra(i)))/
*      (1.d0+1.d1**((objrp(i)-objra(i))))
    Mpt = ran2(raninit)*sumIMF(IMFnum)
    splintcall = 'sumIMF_trace_mass'

    CALL splintcheck(sumIMF,IMFmass,spd2MdsumM2,IMFnum,Mpt,objM
*      ,.false.)

    objmass(i) = objM

end do

CALL cpu_time(time2)
print *, 'Object_generation_done,output_saved_to_',objfilename
print *, 'Time_taken_=_',time2-time1
OPEN(7,file=objfilename)
WRITE(7,1000) '#column1:_dimensionless_galactocentric_distance'
WRITE(7,1001) '#column2:_dimensionless_galactocentric',
*   '_potential'
WRITE(7,1000) '#column3:_dimensionless_object_mass'
WRITE(7,1000) '#column4:_dimensionless_orbital_energy'
WRITE(7,1000) '#column5:_dimensionless_orbital_circularity'
WRITE(7,1000) '#column6:_dimensionless_radial_velocity'
WRITE(7,1000) '#column7:_dimensionless_azimuthal_velocity'
WRITE(7,1000) '#column8:_dimensionless_polar_velocity'
WRITE(7,1000) '#column9:_dimensionless_orbital_pericentre'
WRITE(7,1001) '#column10:_dimensionless_radius_of_circular_',
*   'orbit_with_equal_angular_momentum'
WRITE(7,1001) '#column11:_dimensionless_radius_of_circular_',
*   'orbit_with_equal_energy'
WRITE(7,1000) '#column12:_dimensionless_orbital_apocentre'
WRITE(7,1000) '#column13:_orbital_ellipticity'
do i = 1,objnum
    WRITE(7,1005) objrad(i),objvphi(i),objmass(i),objE(i),objh(i),
*   objvr(i),objvphi(i),objvtheta(i),objrp(i),
*   objrcofL(i),objrcofE(i),objra(i),objellip(i)
end do
CLOSE(7)
end if
STOP
105 FORMAT(2(E50.25))
100 FORMAT(5(E30.20))
101 FORMAT(3(E30.20))
102 FORMAT(E30.20)
112 FORMAT(4(E30.15))
999 FORMAT(5(E30.20,2X),2X,I10)
1000 FORMAT(A)
1001 FORMAT(2A)
1002 FORMAT(10(E30.21,2X))
1003 FORMAT(7(E30.21,2X))
1004 FORMAT(3(E30.20,2X))
1005 FORMAT(13(E30.21,2X))
1006 FORMAT(2(A,E8.3))
1007 FORMAT(2A,E10.3,A,E10.3)
1008 FORMAT(3A)
1010 FORMAT(4(E30.21,2X))
1011 FORMAT(9(E20.10,2X))

```

```

1020 FORMAT(2(E15.6,2X))
1099 FORMAT(2A,1X,1P,E20.10,A)
      END

      ! chops away numeric noise in g(E)
      SUBROUTINE chop(x,y,N,xmin,xmax,newx,newy,numtemp)
      INTEGER N,Nmin,i,imax,imin,ngood,numtemp
      REAL*8 x(N),y(N),xtemp(N),ytemp(N),newx(N),newy(N)
      REAL*8 xmin,xmax,ychop
      CHARACTER*30 splintcall
      COMMON /splintpath/ splintcall
      ychop=0.d0
      Nmin=N/10
      ngood=0
      imin=1
      imax=N

      do i=1,N
        if(y(i).ge.ychop) then
          ngood=ngood+1
          if(ngood.gt.Nmin) then
            imin=i-ngood+1
            goto 10
          end if
        else
          ngood=0
        end if
      end do
10    xmin=x(imin)

      do i=imin,N
        if(y(i).ge.ychop) then
          imax=i
        else
          goto 20
        end if
      end do
20    xmax=x(imax)

      numtemp=0
      do i=imin,imax
        numtemp=numtemp+1
        xtemp(numtemp)=x(i)
        ytemp(numtemp)=y(i)
      end do

      do i =1,numtemp
        newx(i) = xtemp(i)
        newy(i) = ytemp(i)
      end do

      RETURN
      end

      ! log[Phi] as a function of log[r]
      FUNCTION phiofrfn(lgrpt)
      IMPLICIT NONE
      INTEGER intnum,IMFprof,modelprof
      PARAMETER (intnum=1000)
      REAL*8 lgr(intnum),lgphi(intnum),lgrpt,phiofrfn,lgdphidr(intnum)
      REAL*8 spd2rdphi2(intnum),spd2phidr2(intnum),k,p0,u,us,rref
      REAL*8 splgd3phidr3(intnum),phistop,rf
      LOGICAL edgereach
      CHARACTER*30 splintcall
      COMMON /maxpath/ phistop
      COMMON /refpath/ rref,rf
      COMMON /rhopath/ k,p0,edgereach
      COMMON /phiandrp/ lgr,lgphi,spd2rdphi2,spd2phidr2,lgdphidr,
      * splgd3phidr3
      COMMON /modelpath/ modelprof,IMFprof
      COMMON /splintpath/ splintcall
      if (modelprof.eq.4) then ! analytic equation for powerlaw
        if (k.eq.2.d0) then
          phiofrfn = log10(p0*(lgrpt*log(1.d1)-log(rref)))
        else
          phiofrfn = log10(p0*(1.d1**((2.d0-k)*lgrpt)-rref**((2.d0-k))
      * /((3.d0-k)*(2.d0-k)))
        end if
      else if (modelprof.eq.3) then ! analytic equation for dehn
        u = (1.d1**lgrpt)/((1.d1**lgrpt)+1.d0)
        us = rref/(1.d0+rref)
        if (k.eq.2.d0) then
          phiofrfn = log10(4.d0*p0*log(u/us))

```

```

    else
        phiofrfn = log10((p0*2.d0**(4.d0-k)/(3.d0-k))*(us**(2.d0-k))*
            (((u/us)**(2.d0-k))-1.d0)/(2.d0-k))
    *   end if
    else if (modelprof.eq.2) then ! only interpolation for king models
        splintcall = 'lgr_phiofrfn'
        CALL splintcheck(lgr,lgphi,spd2phidr2,intnum,lgrpt,phiofrfn,
    *   .false.)
    else if (modelprof.eq.1) then
        ! powerlaw asymptotic solutions or interpolation for polytropes,
        ! or analytic plummer sphere equations
        if (k.eq.2.5d0) then
            phiofrfn = log10(6.d0*(1.d0-((1.d0+(1.d1**(2.d0*lgrpt))
    *   *(p0/1.8d1)))*(-5.d-1)))
        else
            if (lgrpt.gt.lgr(intnum)) then
                if (k.ne.2.d0) then
                    phiofrfn = log10(2.d0*(k-1.d0))+log10((1.d0-((p0/(2.d0*
    *   (k-1.d0)*(3.d0-k)))*(5.d-1*(2.d0-k)))*
    *   (1.d1**((2.d0-k)*lgrpt)))/(k-2.d0))
                else
                    phiofrfn = log10(log(p0/2.d0)+2.d0*log(1.d1)*lgrpt)
                end if
            else
                splintcall = 'lgr_phiofrfn'
                CALL splintcheck(lgr,lgphi,spd2phidr2,intnum,lgrpt,phiofrfn,
    *   .false.)
            end if
        end if
    end if
    return
END

! log[r] as a function of log[Phi]
FUNCTION rofphifn(lgphipt)
IMPLICIT NONE
INTEGER intnum,IMFprof,modelprof
PARAMETER (intnum=1000)
REAL*8 lgr(intnum),lgphi(intnum),lgphipt,splgd3phidr3(intnum)
REAL*8 spd2rdphi2(intnum),spd2phidr2(intnum),lgdphidr(intnum)
REAL*8 k,p0,us,u,rref,rofphifn,rf
LOGICAL edgereach
CHARACTER*30 splintcall
COMMON /refpath/ rref,rf
COMMON /phiandrp/ lgr,lgphi,spd2rdphi2,spd2phidr2,lgdphidr,
    *   splgd3phidr3
COMMON /modelpath/ modelprof,IMFprof
COMMON /rhopath/ k,p0,edgereach
COMMON /splintpath/ splintcall
if (modelprof.eq.4) then ! analytic equations for powerlaw model
    if (k.eq.2.d0) then
        rofphifn=log10(rref)+(1.d1*lgphipt)/(p0*log(1.d1))
    else
        rofphifn=log10(((1.d1*lgphipt)*(3.d0-k)*(2.d0-k)/p0)
    *   +rref**((2.d0-k))/(2.d0-k))
    *   end if
else if (modelprof.eq.3) then ! analytic equations for dehn model
    us = rref/(rref+1.d0)
    if (k.eq.2.d0) then
        u=us*exp((1.d1*lgphipt)/(4.d0*p0))
    else
        u=(((3.d0-k)*(2.d0-k)*(1.d1*lgphipt)/((2.d0**((4.d0
    *   -k))*p0))+us**((2.d0-k)))*(1.d0/(2.d0-k))
    *   end if
    if (u.gt.1.d0) then
        if (u-1.d0.gt.1.d-4) pause 'dehnen_rofphifn_shafted'
        rofphifn = log10(rf)
    else
        rofphifn = log10(u)-log10(1.d0-u)
    end if
else
    splintcall = 'lgphi_rofphifn'
    CALL splintcheck(lgphi,lgr,spd2rdphi2,intnum,lgphipt,
    *   rofphifn,.false.)
end if
return
10 FORMAT ((E50.25))
END

! returns dlogPhi/dlogr
FUNCTION dphidrofr(lgrpt)
IMPLICIT NONE
INTEGER intnum,modelprof,IMFprof

```



```

REAL*8 PI
PARAMETER (intnum=1000,PI=3.141592654)
REAL*8 dphidrofr, lgr(intnum), lgphi(intnum), lgdphidr(intnum)
REAL*8 spd2phidr2(intnum), splgd3phidr3(intnum), A,B,C
REAL*8 spd2rdphi2(intnum), k, r, p0, lgrpt, phiofrfn, error
LOGICAL edgereach
CHARACTER*30 splntcall

COMMON /phiandrp/ lgr, lgphi, spd2rdphi2, spd2phidr2, lgdphidr,
* splgd3phidr3
COMMON /modelprof/ modelprof, IMFprof
COMMON /rhopath/ k, p0, edgereach
COMMON /splintpath/ splntcall

r = 1.d1**lgrpt
if (modelprof.eq.1) then ! polytropes
  if (lgrpt.lt.lgr(1)) then ! small r asymptotes
    dphidrofr = (p0*(r**2.d0)/3.d0 -
* k*(p0**2.d0)*(r**4.d0)/(6.d1*(k-1.d0)) +
* k*(3.d0*k+1.d1)*(p0**3.d0)*(r**6.d0)/((1008.d1)
* ((k-1.d0)**2.d0)) -
* k*(9.d0*(k**2.d0)+8.6d1*k+2.8d2)*(p0**4.d0)*
* (r**8.d0)/(326592.d1*((k-1.d0)**3.d0)))/
* (1.d1**phiofrfn(lgrpt))
  else if (lgrpt.gt.lgr(intnum)) then ! large r asymptotes
    dphidrofr = ((p0/(3.d0-k))*(5.d-1*(2.d0-k)))*
* ((2.d0*(k-1.d0))*(5.d-1*k))*(r**(2.d0-k))/
* (1.d1**phiofrfn(lgr))
  else ! numerical results for inbetween bits
    splntcall = 'dphidrofr_lgr_lgdphidr'
    CALL splntcheck(lgr, lgdphidr, splgd3phidr3, intnum, lgrpt,
* dphidrofr, .false.)
  end if
else if (modelprof.eq.2) then ! king models
  if (lgrpt.lt.lgr(1)) then ! small r expansions
    A = -(exp(k)*error(k**5.d-1)-2.d0*((k/PI)**5.d-1))/
* (exp(k)*error(k**5.d-1)-2.d0*((k/PI)**5.d-1)*
* (1.d0+2.d0*k/3.d0))

    B = (exp(k)*error(k**5.d-1))/((exp(k)*error(k**5.d-1)
* -2.d0*((k/PI)**5.d-1)*(1.d0+2.d0*k/3.d0))*2.d0)

    C = -(exp(k)*error(k**5.d-1)+((k*PI)*(-5.d-1))-
* ((4.d0*(k**3.d0)*PI)*(-5.d-1)))/((exp(k)*error(k**5.d-1)
* -2.d0*((k/PI)**5.d-1)*(1.d0+2.d0*k/3.d0))*2.4d1)
    ! dphidrofr = p0*(r**2.d0)/(3.d0*(1.d1**phiofrfn(lgrpt)))
    dphidrofr = (p0*(r**2.d0)/3.d0 + (r**4.d0)*(p0**2.d0)*A/3.d1+
* ((A**2.d0)+1.d1*B/3.d0)*(p0**3.d0)*(r**6.d0)/8.4d2
* +((A**3.d0)+5.2d1*A*B/3.d0+7.d1*C/3.d0)*(p0**4.d0)
* (r**8.d0)/4536.d1)/(1.d1**phiofrfn(lgrpt))
  else ! numerical results for everything between small r and r-t
    splntcall = 'dphidrofr_lgr_lgdphidr'
    CALL splntcheck(lgr, lgdphidr, splgd3phidr3, intnum, lgrpt,
* dphidrofr, .false.)
  end if
else if (modelprof.eq.3) then ! dehnen models
  dphidrofr = (2.d0**4.d0-k)*p0*(r**(2.d0-k))*((r+1.d0)**
* (k-3.d0))/((3.d0-k)*(1.d1**phiofrfn(lgrpt)))
else if (modelprof.eq.4) then ! powerlaw models
  dphidrofr = p0*(r**(2.d0-k))/((3.d0-k)*(1.d1**phiofrfn(lgrpt)))
end if
return
END

! returns radius of a circular orbit with equal energy
FUNCTION rEfind(E)
IMPLICIT NONE
INTEGER intnum
PARAMETER (intnum=1000)
REAL*8 spd2LcdEc2(intnum), lgLc(intnum), lgEc(intnum)
REAL*8 lgrc(intnum), spd2rcdEc2(intnum), spd2rcdLc2(intnum)
REAL*8 E, rEfind, Ept
CHARACTER*30 splntcall
COMMON /Ecpath/ lgrc, spd2rcdEc2, spd2rcdLc2
COMMON /Lcpath/ spd2LcdEc2, lgLc, lgEc
COMMON /splintpath/ splntcall
Ept = log10(E)
splntcall = 'lgEc_rEfind'
CALL splntcheck(lgEc, lgrc, spd2rcdEc2, intnum, Ept, rEfind, .false.)
rEfind = 1.d1**rEfind
return
END

```

```

! returns radius of a circular orbit with equal angular mometnum
FUNCTION rLfind(L)
IMPLICIT NONE
INTEGER intnum
PARAMETER (intnum=1000)
REAL*8 spd2LcdEc2(intnum),lgLc(intnum),lgEc(intnum)
REAL*8 lgrc(intnum),spd2rcdEc2(intnum),spd2rcdLc2(intnum)
REAL*8 L,rLfind,Lpt
CHARACTER*30 splintcall
COMMON /Ecpath/ lgrc,spd2rcdEc2,spd2rcdLc2
COMMON /Lcpath/ spd2LcdEc2,lgLc,lgEc
COMMON /splintpath/ splintcall
Lpt = log10(L)
splintcall = 'lgLc_rLfind'
CALL splintcheck(lgLc,lgrc,spd2rcdLc2,intnum,Lpt,rLfind,.false.)
rLfind = 1.d1**rLfind
return
END

! returns g(E)
FUNCTION gofE(lgEpt)
IMPLICIT NONE
INTEGER voltN,i
PARAMETER (voltN=1000)
REAL*8 lgtrapg(voltN),Evec(voltN),spdlld2gdE2(voltN),lgEpt,gofE
CHARACTER*30 splintcall
COMMON /vdpath/ lgtrapg,spdlld2gdE2,Evec
COMMON /splintpath/ splintcall
splintcall = 'Evec_gofE'
CALL splintcheck(Evec,lgtrapg,spdlld2gdE2,voltN,lgEpt,gofE,.true.)
gofE = 1.d1**gofE
return
END

! returns isotropic distribution function
FUNCTION isodf(lgEpt)
IMPLICIT NONE
INTEGER KMAXX,NMAX
PARAMETER (KMAXX=10000,NMAX=50)
INTEGER modelprof,IMFprof,kmax,kount,hount,nvar,nok,nbad
REAL*8 isodf,E,k,p0,dxsav,rref,PI,error,gammaln,xi,xf,eps,h1
REAL*8 hadv,ystart(NMAX),xp(KMAXX),yp(NMAX,KMAXX),lgEpt,rf
LOGICAL edgereach
PARAMETER (PI=3.141592654d0)
COMMON /path/ kmax,kount,dxsav,xp,yp,hount
COMMON /modelpath/ modelprof,IMFprof
COMMON /rhopath/ k,p0,edgereach
COMMON /refpath/ rref,rf
COMMON /isodfpath/ xi
EXTERNAL rkqs,Gne2der,Geq2der
E = 1.d1**lgEpt
if (modelprof.eq.1) then
  if (k.eq.2.d0) then
    isodf = -1.5d0*log10(2.d0*PI)-E/log(1.d1)
  else
    isodf = (k-6.d0)/(4.d0-2.d0*k)
    isodf = isodf*log10(1.d0+E*(2.d0-k)/(2.d0*k-2.d0))
    if (k.lt.2.d0) then
      isodf = isodf+log10(2.d0*k)-2.d0*log10(2.d0-k)
      isodf = isodf+1.5d0*(log10((2.d0-k)/(4.d0*PI*(k-1.d0))))
      isodf = isodf+(gammaln(k/(2.d0-k)+1.5d0)-gammaln(k/(2.d0-k)+2.d0))
      * /log(1.d1)
    else if (k.gt.2.d0) then
      isodf = isodf+log10(2.d0*k)-2.d0*log10(k-2.d0)
      isodf = isodf+1.5d0*(log10((k-2.d0)/(4.d0*PI*(k-1.d0))))
      isodf = isodf+(gammaln(k/(k-2.d0)-1.d0)-gammaln(k/(k-2.d0)-5.d-1))
      * /log(1.d1)
    end if
  end if
else if (modelprof.eq.2) then
  isodf = log10(exp(k-E)-1.d0)-1.5d0*log10(2.d0*PI)
  isodf = isodf-log10(exp(k)*error(k**5.d-1)-
    * ((4.d0*k/PI)**5.d-1)*(1.d0+2.d0*k/3.d0))
else if (modelprof.eq.3) then
  nvar = 1
  kmax = 0
  eps=1.d-7
  hl=1.d-10
  hadv=1.d-10
  if (k.ne.2.d0) then
    xi = (E)*(3.d0-k)*(2.d0-k)/(p0*(2.d0**((4.d0-k))))
    xi = (xi+((rref/(rref+1.d0))*(2.d0-k)))*(1.d0/(2.d0-k))
    xf = 1.d0
  end if
end if

```

```

        ystart(1) = 0.d0
        CALL odeint(ystart,nvar,xi+5.d-16,xf,eps,h1,hadv,nok,nbad,
*                               Gne2der,rkqs)
        isodf = 1.5d0*log10((3.d0-k)/p0)+(5.d-1*k-2.5d0)*log10(2.d0)
        isodf = isodf-2.d0*log10(PI)+log10(ystart(1))
        else if (k.eq.2.d0) then
            xi = (rref/(rref+1.d0))*exp((E)/(4.d0*p0))
            xf = 1.d0
            ystart(1) = 0.d0
            CALL odeint(ystart,nvar,xi+1.d-15,xf,eps,h1,hadv,nok,nbad,
*                               Geq2der,rkqs)
            isodf = log10(ystart(1))-5.d-1*log10(2.d0)
            isodf = isodf-2.d0*log10(PI)-1.5d0*log10(p0)
        end if
        else if (modelprof.eq.4) then
            if (k.lt.2.d0) then
                isodf = log10(k)-5.d-1*log10(2.d0*(2.d0-k))
                isodf = isodf+1.5d0*log10((3.d0-k)/(p0*PI))
                isodf = isodf+(gammaln(k/(2.d0-k)+1.5d0)-gammaln(k/(2.d0-k)+2.d0))
*                               /log(1.d1)
            else if (k.gt.2.d0) then
                isodf=log10((3.d0-k)*(2.d0-k)*(E)/p0+rref*(2.d0-k))
                log10((3.d0-k)*(2.d0-k)*(E)/p0+rref*(2.d0-k))
            else if (k.gt.2.d0) then
                isodf = log10(k)-5.d-1*log10(2.d0*(k-2.d0))
                isodf = isodf+1.5d0*log10((3.d0-k)/(p0*PI))
                isodf = isodf+(gammaln(k/(k-2.d0)-1.d0)-gammaln(k/(k-2.d0)-5.d-1))
*                               /log(1.d1)
            else if (k.eq.2.d0) then
                isodf=log10(rref*(2.d0-k)-(3.d0-k)*(k-2.d0)*(E)/p0)
                log10(rref*(2.d0-k)-(3.d0-k)*(k-2.d0)*(E)/p0)
            else if (k.eq.2.d0) then
                isodf = -2.d0*log10(rref)-1.5d0*log10(p0*PI)
                isodf = isodf-2.d0*(E)/(p0*log(1.d1))
            end if
        end if
        return
    END

! system of ODEs for Dehnen models
SUBROUTINE Gne2der(x,y,dydx)
    IMPLICIT NONE
    REAL*8 x,y(*),dydx(*),k,p0,xi
    LOGICAL edgereach
    COMMON /rhopath/ k,p0,edgereach
    COMMON /isodfpath/ xi
    dydx(1) = (4.d0-k)*(x**4.d0)+2.d0*(k-3.d0)*(x**3.d0)
    dydx(1) = (dydx(1)+2.d0*(1.d0-k)*x+k)/(x**3.d0)
    dydx(1) = dydx(1)/((((x**(2.d0-k))-(xi**(2.d0-k)))/(2.d0-k))
*                               **5.d-1)
    return
END

! system of ODEs for Jaffe model
SUBROUTINE Geq2der(x,y,dydx)
    IMPLICIT NONE
    REAL*8 x,y(*),dydx(*),k,p0,xi
    LOGICAL edgereach
    COMMON /rhopath/ k,p0,edgereach
    COMMON /isodfpath/ xi
    dydx(1) = ((x**4.d0)-(x**3.d0)-x+1.d0)
    dydx(1) = dydx(1)/((x**3.d0)*((log(x/xi))**5.d-1))
    return
END

! system of ODEs for number interior to radius
SUBROUTINE traceint(x,y,dydx)
    IMPLICIT NONE
    REAL*8 x,y(*),dydx(*),PI,tracerhofn
    PARAMETER (PI=3.141592654d0)
    if (x.eq.0.d0) then
        dydx(1) = 0.d0
    else
        dydx(1) = 4.d0*PI*(x**2.d0)*tracerhofn(x)
    end if
    return
END

! system of ODEs for tracer number density
SUBROUTINE dfrho(x,y,dydx)
    IMPLICIT NONE
    INTEGER nrep
    REAL*8 x,y(*),dydx(*),intphipt,gofE,H,upplim
    LOGICAL nonint
    COMMON /gintpath/ intphipt

```

```

COMMON /nonintpath/ nrep,nonint,upplim
dydx(1) = gofE(x)*H(x,intphipt)
if (dydx(1).eq.0.d0) then ! for some models, phi turns very flat
                           ! in the outer regions. with numerical
                           ! roundoff, this can make x and
                           ! intphipt appear the same, making H=0.
                           ! odeint will then happily chug along
                           ! without ever getting anywhere, so
                           ! check for this happening and abort
                           ! integration when it is detected.

  nrep = nrep + 1
  if (nrep.ge.1000) then
    nonint = .true. ! set detection flag
    x = upplim! make odeint think upper limit has been reached
  end if
end if
return
END

! tracer number density after accounting for Emax cutoff
FUNCTION tracerhofn(r)
IMPLICIT NONE
INTEGER tracedatnum
PARAMETER (tracedatnum=1000)
REAL*8 rhoht, trcelgrho(tracedatnum),r,rpt
REAL*8 trcelgr(tracedatnum),spd2rhodr2(tracedatnum),tracerhofn
CHARACTER*30 splintcall
COMMON /tracepath/ trcelgrho, trcelgr, spd2rhodr2
COMMON /splintpath/ splintcall
splintcall = 'trcelgr-tracerhofn'
rpt = log10(r)
CALL splintcheck(trcelgr, trcelgrho, spd2rhodr2, tracedatnum,
*               rpt, rhoht, .false.)
tracerhofn = 1.d1**rhoht
return
END

! returns tracer number density before accounting for Emax cutoff
FUNCTION numdenstrace(r)
IMPLICIT NONE
COMMON /tracepath/ traceinpl, traceoupl, tracetrans, traceprof
INTEGER traceprof
REAL*8 numdenstrace, lgrhofn, r, phiofrfn, traceinpl, traceoupl
REAL*8 tracetrans, tracenu
if (traceprof.eq.1) then ! self consistent
  numdenstrace = 1.d1**lgrhofn(phiofrfn(log10(r)), log10(r))
end if
if (traceprof.eq.2) then ! double power law
  numdenstrace = tracenu(traceinpl, traceoupl, tracetrans, r)
end if
return
END

! double power law
FUNCTION tracenu(gamma,beta,alpha,r)
IMPLICIT NONE
REAL*8 tracenu,gamma,beta,alpha,r
tracenu = (r**(-gamma))*
* ((1.d0+r**(1.d0/alpha))**((gamma-beta)*alpha))
return
END

! system of ODEs for Jeans equation in logarithmic form
SUBROUTINE jeans(x,y,dydx)
IMPLICIT NONE
INTEGER intnum
PARAMETER (intnum=1000)
REAL*8 x,y(*),dydx(*),lgrhofn,phiofrfn,dphidrofr
dydx(1)=(1.d1**((phiofrfn(x)+lgrhofn(phiofrfn(x),x)))*dphidrofr(x)
return
END

! system of ODEs for Poissons equation in logarithmic form
SUBROUTINE poisson(x,y,dydx)
IMPLICIT NONE
REAL*8 x,y(*),dydx(*),lgrhofn,p0,k
LOGICAL edgereach
COMMON /rhopath/ k,p0,edgereach
dydx(1) = y(2)
dydx(2) = p0*(1.d1**((2.d0*x-y(1)+lgrhofn(y(1),x))))
dydx(2) = (dydx(2)-y(2)*(1.d0+y(2)))*log(1.d1)
return
END

```

```

! calculates dimensionless density as a function of dimensionless potential
Function lgrhofn(lgphi, lgr)
IMPLICIT NONE
INTEGER IMFprof, modelprof
REAL*8 lgrhofn, polyrho, kingrho, dehnrho, powerrho, lgphi, lgr
COMMON /modelpath/ modelprof, IMFprof
if (modelprof.eq.1) lgrhofn = polyrho(lgphi, lgr)
if (modelprof.eq.2) lgrhofn = kingrho(lgphi, lgr)
if (modelprof.eq.3) lgrhofn = dehnrho(lgphi, lgr)
if (modelprof.eq.4) lgrhofn = powerrho(lgphi, lgr)
return
END

! analytic equation for dehn model density as a function of radius
Function dehnrho(lgphi, lgr)
IMPLICIT NONE
REAL*8 dehnrho, lgphi, k, p0, lgr
LOGICAL edgereach
COMMON /rhopath/ k, p0, edgereach
dehnrho = (4.d0-k)*(log10(2.d0)-log10((1.d1**lgr)+1.d0))-k*lgr
return
END

! analytic equation for power law density as a function of radius
Function powerrho(lgphi, lgr)
IMPLICIT NONE
REAL*8 powerrho, lgphi, k, p0, lgr
LOGICAL edgereach
COMMON /rhopath/ k, p0, edgereach
powerrho = -k*lgr
return
END

! analytic equation for polytrope density as a function of potential
Function polyrho(lgphi, lgr)
Implicit None
REAL*8 polyrho, lgphi, k, p0, lgr
LOGICAL edgereach
COMMON /rhopath/ k, p0, edgereach
if (k.eq.2.d0) then
    polyrho = -(1.d1**lgphi)/log(1.d1)
else
    if (k.eq.2.5d0) then
        polyrho = -2.5d0*log10((p0/1.8d1)*(1.d1**lgr))+1.d0
    else
        polyrho = (k/(k-2.d0))*log10(1.d0-(1.d1**lgphi)*(k-2.d0)/
                                     (2.d0*(k-1.d0)))
    end if
end if
return
END

! analytic(ish) equation for king model density as a function of potential
FUNCTION kingrho(lgphi, lgr)
IMPLICIT NONE
REAL*8 kingrho, lgphi, k, PI, W, error, a, b, p0, lgr
PARAMETER (PI=3.141592654d0)
LOGICAL edgereach
COMMON /rhopath/ k, p0, edgereach
W = k-(1.d1**lgphi)
if (W.gt.0.d0) then
    a = log10(exp(W)*error(sqrt(W))-sqrt(4.d0*W/PI)*(1.d0+2.d0*W/
                                                         3.d0))
    b = log10(exp(k)*error(sqrt(k))-sqrt(4.d0*k/PI)*(1.d0+2.d0*k/
                                                         3.d0))
    kingrho = a-b
else
    kingrho = log10(0.d0)
end if
return
END

! system of ODEs for integrating mass function
SUBROUTINE IMFint(x, y, dydx)
IMPLICIT NONE
REAL*8 x, y(*), dydx(*), IMF, M
M = 1.d1**x
dydx(1) = IMF(x)
return
END

! returns dN/dlogM at log[M] for selected mass function

```

```

FUNCTION IMF(lgM)
IMPLICIT NONE
INTEGER IMFprof,modelprof
REAL*8 IMF,lgM,Schechter,Gaussian
COMMON /modelpath/ modelprof,IMFprof
if (IMFprof.eq.1) IMF = Schechter(lgM)
if (IMFprof.eq.2) IMF = Gaussian(lgM)
if (IMFprof.eq.3) IMF = Schechter(lgM)
return
END

! Schechter mass function
FUNCTION Schechter(lgM)
IMPLICIT NONE
REAL*8 Schechter,M,IMFparam1,IMFparam2,lgM
COMMON /IMFpath/ IMFparam1,IMFparam2
M = 1.d1**lgM
Schechter = (1.d1**((1.d0-IMFparam2)*lgM))*exp(-M/IMFparam1)
return
END

! lognormal mass function
FUNCTION Gaussian(lgM)
IMPLICIT NONE
REAL*8 Gaussian,lgM,IMFparam1,IMFparam2
COMMON /IMFpath/ IMFparam1,IMFparam2
Gaussian = exp(-5.d-1*((lgM-IMFparam1)/IMFparam2)**2.d0))
return
END

! provides the function to be bisected when solving for rp and ra
FUNCTION hfunc(rturn)
IMPLICIT NONE
REAL*8 hfunc,rturn,phiofrfn,E,L,lgrE,rmaxpt,LcplusL0
COMMON /hfuncpath/ lgrE,rmaxpt
COMMON /ELpath/ E,L
if (rturn.eq.lgrE) then ! by definition, (E-phi(rE))rE2 = vc2(rE)rE2 = Lc(E)2
hfunc = (LcplusL0(E)**2.d0)-(L**2.d0)
else if (rturn.eq.rmaxpt) then ! by definition, (E-phi(rmaxpt)) = 0
hfunc = -L**2.d0
else
hfunc = 2.d0*(1.d1**E-1.d1**phiofrfn(rturn))*
* (1.d1**(2.d0*rturn))-(L**2.d0)
end if
return
END

! solves the Volterra equation and returns g(E)
SUBROUTINE voltrap(vecN,abscvec,h,g,f,K)
IMPLICIT NONE
INTEGER vecN,i,n
REAL*8 h(vecN-1),g(vecN),f,K,absc(vecN),sumA,sumB,abscvec(vecN)
do n = 1,vecN
absc(n) = 1.d1**(abscvec(n)) ! define logarithmically spaced x points
end do
g(vecN) = 2.d0*f(absc(vecN-1))/
* (h(vecN-1)*K(abscvec(vecN),abscvec(vecN-1)))
do n = vecN-1,2,-1
sumA = 2.d0*f(absc(n-1))
sumA =sumA-g(vecN)*K(abscvec(vecN),abscvec(n-1))*h(vecN-1)
sumB =0.d0
do i =n+1,vecN-1
sumB = sumB + g(i)*K(abscvec(i),abscvec(n-1))*(h(i)+h(i-1))
end do
g(n) = sumA - sumB
g(n) = g(n)/((h(n)+h(n-1))*K(abscvec(n),abscvec(n-1)))
end do
return
END

! returns rho*r/4pi
FUNCTION voltf(phipt)
IMPLICIT NONE
REAL*8 numdenstrace,phipt,voltf,rofphifn,PI,rpt,phi
PARAMETER (PI=3.141592654d0)
phi = phipt
rpt = 1.d1**rofphifn(log10(phi))
voltf = numdenstrace(rpt)*rpt/(4.d0*PI*log(1.d1))
return
END

! returns Kernel of distribution function integrand
FUNCTION H(lgEpt,lgphipt)

```

```

IMPLICIT NONE
INTEGER KMAXX,NMAX
PARAMETER (KMAXX=10000,NMAX=50)
INTEGER nvar,kmax2,nok,nbad,kount2,hount2,jtype
REAL*8 H,hmax,lgEpt,lgphipt,eps,h1,hadv,ystart(NMAX),xi,xf
REAL*8 xp2(KMAXX),yp2(NMAX,KMAXX),dxsav2,ja,jb,LcplusL0
REAL*8 beta
COMMON /Hpath/ xf
COMMON /path2/ kmax2,kount2,dxsav2,xp2,yp2,hount2
COMMON /jpath/ ja,jb,jtype
EXTERNAL jint,rkqs2
if (lgEpt.eq.lgphipt) then
  H = 0.d0
else
  eps=1.d-5
  h1=1.d-10
  hadv=1.d-10
  nvar = 1
  kmax2 = 0 ! = 100
  ystart(1)=0.d0
  xi = 0.d0
  xf = hmax(lgEpt,lgphipt)
  if (jtype.eq.1) then
    ystart(1) = 5.d-1*(xf**(1.d0-2.d0*ja))*beta(5.d-1,1.d0-ja)
  else
    CALL odeint2(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,jint,
*                                     rkqs2)
  end if
  H = (1.d1**lgEpt)*ystart(1)*LcplusL0(lgEpt)
end if
return
END

! returns hmax
FUNCTION hmax(lgEpt,lgphipt)
IMPLICIT NONE
REAL*8 lgEpt,lgphipt,hmax,LcplusL0,rofphifn
hmax =5.d-1*log10(1.d0-1.d1**(lgphipt-lgEpt))+rofphifn(lgphipt)
* +5.d-1*log10(2.d0)+5.d-1*lgEpt-log10(LcplusL0(lgEpt))
hmax = 1.d1**hmax
end if
return
END

! returns angular momentum of a circular orbit with energy E
FUNCTION LcofE(lgEpt)
IMPLICIT NONE
INTEGER intnum
PARAMETER (intnum=1000)
REAL*8 LcofE,lgEc(intnum),spd2LcdEc2(intnum),lgLc(intnum)
REAL*8 lgEpt,lgR(intnum),lgphi(intnum),spd2rdphi2(intnum)
REAL*8 spd2phidr2(intnum),lgdphidr(intnum),splgd3phidr3(intnum)
CHARACTER*30 splntcall
COMMON /Lcpath/ spd2LcdEc2,lgLc,lgEc
COMMON /splintpath/ splntcall
COMMON /phiandrpah/ lgr,lgphi,spd2rdphi2,spd2phidr2,lgdphidr,
* splgd3phidr3

splntcall = 'lgEc_LcofE'

CALL splntcheck(lgEc,lgLc,spd2LcdEc2,intnum,lgEpt,LcofE,
*                                     .false.)
LcofE = 1.d1**LcofE
return
END

! returns angular momentum of a circular orbit with energy E
! plus anisotropy parameter L0
FUNCTION LcplusL0(lgEpt)
IMPLICIT NONE
REAL*8 LcplusL0,L0,lgEpt,LcofE
COMMON /L0path/ L0
LcplusL0 = LcofE(lgEpt)+L0
return
END

! system of ODEs for integrating over j
SUBROUTINE jint(x,y,dydx)
IMPLICIT NONE
REAL*8 x,y(*),dydx(*),j,h,xf
COMMON /Hpath/ xf
h = sqrt(xf**2.d0-x**2.d0)
dydx(1)=j(h)

```

```

return
END

FUNCTION jfirst(h,ja)
IMPLICIT NONE
REAL*8 jfirst,h,ja
jfirst = h**(-2.d0*ja)
return
END

FUNCTION jsecond(h,ja)
IMPLICIT NONE
REAL*8 jsecond,h,ja
jsecond = exp(-ja*h)
return
END

FUNCTION jthird(h,ja,jb)
IMPLICIT NONE
REAL*8 jthird,h,ja,jb
jthird = exp(-5.d-1*(((h-ja)/jb)**2.d0))
return
END

FUNCTION jfourth(h,ja)
IMPLICIT NONE
REAL*8 jfourth,h,ja
jfourth = 1.d0-(h**ja)*(1.d0-ja*log(h+1.d-10))
return
END

FUNCTION jfifth(h,ja)
IMPLICIT NONE
REAL*8 jfifth,h,ja
jfifth = 5.d-1*(1.d0+h*sin(ja*h))
return
END

! circularity functions
FUNCTION j(h)
IMPLICIT NONE
INTEGER jtype
REAL*8 j,h,ja,jb,jfirst,jsecond,jthird,jfourth,jfifth
COMMON /jpath/ ja,jb,jtype
if (jtype.eq.1) j=jfirst(h,ja)
if (jtype.eq.2) j=jsecond(h,ja)
if (jtype.eq.3) j=jthird(h,ja,jb)
if (jtype.eq.4) j=jfourth(h,ja)
if (jtype.eq.5) j=jfifth(h,ja)
if (j.ne.j.or.1.d0/j.eq.0.d0) then
  print *, 'h=',h,'_j=',j
  pause 'j_NAN_or_INF'
end if
if (j.gt.1.d0.or.j.lt.0.d0) then
  print *, 'h=',h,'_j=',j
  pause 'j_must_be_bounded_by_0<=j<=1'
end if
return
END

! system of ODEs for the differential energy distribution
SUBROUTINE DEDint(x,y,dydx)
IMPLICIT NONE
INTEGER KMAXX,NMAX,DEDN
PARAMETER (KMAXX=10000,NMAX=50,DEDN=1000)
INTEGER kmax2,nvar,nok,nbad,kount2,hount2,jtype
REAL*8 dydx(*),y(*),x,eps,h1,hadv,ystart(NMAX),phiofrfn
REAL*8 xi,xf,hmax,dxsav2,yp2(KMAXX),yp2(NMAX,KMAXX),Ept
REAL*8 spd2DEDdE2(DEDN),DED(DEDN),DEDEvec(DEDN)
REAL*8 ja,jb
COMMON /Hpath/ xf
COMMON /DEDpath/ Ept,spd2DEDdE2,DED,DEDEvec
COMMON /path2/ kmax2,kount2,dxsav2,yp2,yp2,hount2
COMMON /jpath/ ja,jb,jtype
EXTERNAL jint,rkqs2
kmax2 = 0 ! = 100
eps=1.d-7
h1=1.d-10
hadv=1.d-10
nvar = 1
ystart(1)=0.d0
xi = 0.d0
xf = hmax(Ept,phiofrfn(log10(x)))

```



```

if (jtype.eq.1) then
  ystart(1) = 5.d-1*(xf**(1.d0-2.d0*ja))*beta(5.d-1,1.d0-ja)
else
  CALL odeint2(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,jint,rkqs2)
end if
dydx(1)=x*ystart(1)
return
END

! system of ODEs for integrating over the differential energy distribution
SUBROUTINE sumDEDint(x,y,dydx)
IMPLICIT NONE
INTEGER DEDN
PARAMETER (DEDN=1000)
REAL*8 dydx(*),y(*),x,Zpt
REAL*8 spd2DEDdE2(DEDN),DED(DEDN),Ept,DEDEvec(DEDN)
CHARACTER*30 splintcall
COMMON /DEDpath/ Ept,spd2DEDdE2,DED,DEDEvec
splintcall = 'DEDEvec_sumDEDint'
CALL splintcheck(DEDEvec,DED,spd2DEDdE2,DEDN,x,Zpt,.false.)
dydx(1) = Zpt
return
END

! system of ODEs for radial velocity dispersions
SUBROUTINE dfvdr2(x,y,dydx)
IMPLICIT NONE
INTEGER KMAXX,NMAX
PARAMETER (KMAXX=10000,NMAX=50)
INTEGER kmax2,nvar,nok,nbad,kount2,hount2,jtype
REAL*8 dydx(*),y(*),x,lgphipt,hmaxval,eps,h1,hadv,ystart(NMAX)
REAL*8 xi,xf,LcplusL0,hmax,gofE,dxsav2,yp2(KMAXX),yp2(NMAX,KMAXX)
REAL*8 ja,jb,E,beta
COMMON /vdHpath/ hmaxval
COMMON /gintpath/ lgphipt
COMMON /path2/ kmax2,kount2,dxsav2,yp2,yp2,hount2
COMMON /jpath/ ja,jb,jtype
EXTERNAL Hr,rkqs2
E = 1.d1**x
hmaxval = hmax(x,lgphipt)
kmax2 = 0 ! = 100
eps=1.d-5
h1=1.d-10
hadv=1.d-10
nvar = 1
ystart(1)=0.d0
xi = 0.d0
xf = hmaxval
if (jtype.eq.1) then
  ystart(1) = 5.d-1*(xf**(3.d0-2.d0*ja))*beta(1.5d0,1.d0-ja)
else
  CALL odeint2(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,Hr,rkqs2)
end if
dydx(1)=E*gofE(x)*(LcplusL0(x)**3.d0)*ystart(1)
return
END

! system of ODEs for inner integral of radial velocity dispersions
SUBROUTINE Hr(x,y,dydx)
IMPLICIT NONE
REAL*8 x,y(*),dydx(*),j,hmaxval
COMMON /vdHpath/ hmaxval
dydx(1) = x*((hmaxval**2.d0)-(x**2.d0))**5.d-1*j(x)
return
END

! system of ODEs for tangential velocity dispersions
SUBROUTINE dfvdt2(x,y,dydx)
IMPLICIT NONE
INTEGER KMAXX,NMAX
PARAMETER (KMAXX=10000,NMAX=50)
INTEGER kmax2,nvar,nok,nbad,kount2,hount2,jtype
REAL*8 dydx(*),y(*),x,lgphipt,hmaxval,eps,h1,hadv,ystart(NMAX)
REAL*8 xi,xf,LcplusL0,hmax,gofE,dxsav2,yp2(KMAXX),yp2(NMAX,KMAXX)
REAL*8 ja,jb,E,beta
COMMON /vdHpath/ hmaxval
COMMON /gintpath/ lgphipt
COMMON /path2/ kmax2,kount2,dxsav2,yp2,yp2,hount2
COMMON /jpath/ ja,jb,jtype
EXTERNAL Ht,rkqs2
E = 1.d1**x
hmaxval = hmax(x,lgphipt)
kmax2 = 0 ! = 100

```

```

eps=1.d-5
h1=1.d-10
hadv=1.d-10
nvar = 1
ystart(1)=0.d0
xi = 0.d0
xf = hmaxval
if (jtype.eq.1) then
  ystart(1) = 5.d-1*(xf**(3.d0-2.d0*ja))*beta(5.d-1,2.d0-ja)
else
  CALL odeint2(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,Ht,rkqs2)
end if
dydx(1)=E*gofE(x)*(LcplusL0(x)**3.d0)*ystart(1)
return
END

! system of ODEs for inner integral of tangential velocity dispersions
SUBROUTINE Ht(x,y,dydx)
IMPLICIT NONE
REAL*8 x,y(*),dydx(*),h,j,hmaxval
COMMON /vdHpath/ hmaxval
h = ((hmaxval**2.d0)-(x**2.d0))*5.d-1
dydx(1) = (h**2.d0)*j(h)
return
END

! system of ODEs for E sampling
SUBROUTINE Eint(x,y,dydx)
IMPLICIT NONE
INTEGER KMAXX,NMAX
PARAMETER (KMAXX=10000,NMAX=50)
INTEGER kmax2,nvar,kount2,hount2,nok,nbad,jtype
REAL*8 x,y(*),dydx(*),gofE,LcplusL0,dxsav2,yp2(KMAXX)
REAL*8 ystart(NMAX),yp2(NMAX,KMAXX),hmaxval,intphipt,E,eps,h1
REAL*8 hadv,xi,xf,hmax,ja,jb,u,beta
COMMON /jpath/ ja,jb,jtype
COMMON /gintpath/ intphipt
COMMON /path2/ kmax2,kount2,dxsav2,yp2,yp2,hount2
COMMON /Hpath/ hmaxval
EXTERNAL jint,rkqs2
E = 1.d1**x
hmaxval = hmax(x,intphipt)
kmax2 = 0 ! = 100
eps=1.d-5
h1=1.d-10
hadv=1.d-10
nvar = 1
ystart(1)=0.d0
xi = 0.d0
xf = hmaxval
if (jtype.eq.1) then
  ystart(1) = 5.d-1*(xf**(1.d0-2.d0*ja))*beta(5.d-1,1.d0-ja)
else
  CALL odeint2(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,jint,rkqs2)
end if
dydx(1) = gofE(x)*LcplusL0(x)*E*ystart(1)
return
END

! system of ODEs h sampling
SUBROUTINE hint(x,y,dydx)
IMPLICIT NONE
REAL*8 x,y(*),dydx(*),ja,jb
INTEGER jtype
COMMON /jpath/ ja,jb,jtype
dydx(1) = (x**(-ja))*((1.d0-x)**(-5.d-1))
return
END

! slightly modifies values to force monotonicity
SUBROUTINE monotonic(x1,n,x2,h)
IMPLICIT NONE
INTEGER n,i
REAL*8 x1(n),x2(n),h
x2(1)=x1(1)
do i = 1,n-1
  x2(i+1)=x1(i+1)
  if ((1.d0+h)*x2(i).gt.x2(i+1)) then
    x2(i+1) = (1.d0+h)*x2(i)
  end if
end do
return
END

```

```

SUBROUTINE splintcheck(xa,ya,y2a,n,x,y,linintrp)  ! checks for extrapolation
                                                    ! before a splint call

IMPLICIT NONE
INTEGER n,splinterrn
REAL*8 xa(n),ya(n),y2a(n),x,y,splintdelta
LOGICAL linintrp
PARAMETER (splintdelta=1.d-8)
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /errpath/ splinterrn
CALL splint(xa,ya,y2a,n,x,y,linintrp)
if (x.lt.xa(1)) then
    write(31,11) 'below',x,xa(1),xa(1)-x,splintcall
    splinterrn = splinterrn + 1
else if (x.gt.xa(n)) then
    write(31,11) 'above',x,xa(n),xa(n)-x,splintcall
    ! count extrapolation errors and print information to a file for later inspection
    splinterrn = splinterrn + 1
end if
return
11 FORMAT(A,2X,3(E30.15,2X),A)
END

! the following subroutine for interpolating over an array
! is slightly modified from that in Press et al. 1992
SUBROUTINE splint(xa,ya,y2a,n,x,y,linintrp)
IMPLICIT NONE
INTEGER n
REAL*8 x,y,xa(n),y2a(n),ya(n)
INTEGER k,khi,klo
REAL*8 a,b,h
CHARACTER*30 splintcall
LOGICAL linintrp,linin
COMMON /linpath/ linin
COMMON /splintpath/ splintcall
klo=1
khi=n
1  if (khi-klo.gt.1) then
    k=(khi+klo)/2
    if (xa(k).gt.x) then
        khi=k
    else
        klo=k
    endif
    goto 1
endif
h=xa(khi)-xa(klo)
if (h.eq.0.d0) then
    pause 'bad_xa_input_in_splint'
end if
a=(xa(khi)-x)/h
b=(x-xa(klo))/h
y=a*ya(klo)+b*ya(khi)
! if linintrp = true or linin = true, interpolate linearly,
! otherwise do cubic interpolation
if ((linintrp.eqv..false.).and.(linin.eqv..false.)) then
    y=y+((a**3-a)*y2a(klo)+(b**3-b)*y2a(khi))*(h**2)/6.d0
end if
return
END

! a modified version of sort from Press et al. 1992
! sorts multiple arrays simultaneously
SUBROUTINE multisort(numvec,n,sort1,sort2,sort3,sort4,sort5,sort6
*,sort7,sort8)
implicit none
INTEGER numvec,n,iwksp(n)
REAL*8 sort1(n),sort2(n),sort3(n),sort4(n),sort5(n)
REAL*8 sort6(n),sort7(n),sort8(n),WKSP(n)
INTEGER j,numdone
numdone=0
call indexx(n,sort1,iwksp)
do 11 j=1,n
    wksp(j)=sort1(j)
11 continue
do 12 j=1,n
    sort1(j)=wksp(iwksp(j))
12 continue
numdone=numdone+1
if (numdone.eq.numvec) RETURN
do 13 j=1,n
    wksp(j)=sort2(j)

```

```

13      continue
      do 14 j=1,n
         sort2(j)=wksp(iwksp(j))
14      continue
         numdone=numdone+1
         if(numdone.eq.numvec) RETURN
         do 15 j=1,n
            wksp(j)=sort3(j)
15      continue
         do 16 j=1,n
            sort3(j)=wksp(iwksp(j))
16      continue
         numdone=numdone+1
         if(numdone.eq.numvec) RETURN
         do 17 j=1,n
            wksp(j)=sort4(j)
17      continue
         do 18 j=1,n
            sort4(j)=wksp(iwksp(j))
18      continue
         numdone=numdone+1
         if(numdone.eq.numvec) RETURN
         do 19 j=1,n
            wksp(j)=sort5(j)
19      continue
         do 20 j=1,n
            sort5(j)=wksp(iwksp(j))
20      continue
         numdone=numdone+1
         if(numdone.eq.numvec) RETURN
         do 21 j=1,n
            wksp(j)=sort6(j)
21      continue
         do 22 j=1,n
            sort6(j)=wksp(iwksp(j))
22      continue
         numdone=numdone+1
         if(numdone.eq.numvec) RETURN
         do 23 j=1,n
            wksp(j)=sort7(j)
23      continue
         do 24 j=1,n
            sort7(j)=wksp(iwksp(j))
24      continue
         numdone=numdone+1
         if(numdone.eq.numvec) RETURN
         do 25 j=1,n
            wksp(j)=sort8(j)
25      continue
         do 26 j=1,n
            sort8(j)=wksp(iwksp(j))
26      continue
      RETURN
      end

      ! the following is a slightly modified version of a driver for
      ! Runge-Kutta integration from Press et al. 1992, that forces a
      ! step when a step cannot be taken while satisfying the required eps error
      SUBROUTINE odeint(ystart,nvar,x1,x2,eps,h1,hadv,nok,nbad,derivs,
      *rkqs)
      implicit none
      INTEGER nbad,nok,nvar,KMAXX,MAXSTP,NMAX
      REAL*8 eps,h1,hadv,x1,x2,ystart(nvar),TINY
      EXTERNAL derivs,rkqs
      PARAMETER (MAXSTP=100000000,NMAX=50,KMAXX=10000,TINY=1.d-30)
      INTEGER i,kmax,kount,nstp,hount
      REAL*8 dxsav,h,hdid,hnext,x,xsav,dydx(NMAX),xp(KMAXX),y(NMAX),
      *yp(NMAX,KMAXX),yscal(NMAX)
      COMMON /path/ kmax,kount,dxsav,xp,yp,hount
      x = x1
      h = sign(h1,x2-x1)
      ! align sign of hadv with direction in which integration is proceeding
      hadv = sign(hadv,x2-x1)
      nok = 0
      nbad = 0
      kount = 0
      hount = 0
      do 11 i=1,nvar
         y(i)=ystart(i)
11      continue
      if (kmax.gt.0) xsav=x-2.d0*dxsav
      do 16 nstp=1,MAXSTP
         call derivs(x,y,dydx)

```

```

do 12 i=1,nvar
  yscal(i)=abs(y(i))+abs(h*dydx(i))+TINY
12 continue
  if(kmax.gt.0) then
    if(abs(x-xsav).gt.abs(dxsav)) then
      if(kount.lt.kmax-1)then
        kount=kount+1
        xp(kount)=x
        do 13 i=1,nvar
          yp(i,kount)=y(i)
13        continue
          xsav=x
        endif
      endif
    endif
    if((x+h-x2)*(x+h-x1).gt.0.d0) h=x2-x
    call rkqs(y,dydx,nvar,x,h,hadv,eps,yscal,hdid,hnext,derivs)
    if(hdid.eq.hadv) hount = hount+1 ! track number of forced advances made
    if(hdid.eq.h)then
      nok=nok+1
    else
      nbad=nbad+1
    endif
    if((x-x2)*(x2-x1).ge.0.d0)then
      do 14 i=1,nvar
        ystart(i)=y(i)
14      continue
        if(kmax.ne.0) then
          kount=kount+1
          xp(kount)=x
          do 15 i=1,nvar
            yp(i,kount)=y(i)
15          continue
        endif
      return
    endif
    h=hnext
16 continue
    pause 'too many steps in codeint'
33 return
END

! the following is a slightly modified version of a Runge-Kutta stepper program
! that takes steps whilst monitoring the estimated error. The modification forces
! a step to be taken when rkqs cannot satisfy the error requirement and would
! otherwise stop.
SUBROUTINE rkqs(y,dydx,n,x,htry,hadv,eps,yscal,hdid,hnext,derivs)
  implicit none
  INTEGER n,NMAX
  REAL*8 eps,hdid,hnext,htry,x,dydx(n),y(n),yscal(n),hadv
  EXTERNAL derivs
  PARAMETER (NMAX=50)
CU  USES derivs,rkck
  INTEGER i
  REAL*8 errmax,h,htemp,xnew,yerr(NMAX),ytemp(NMAX),SAFETY,PGROW,
  *PSHRNK,ERRCON
  PARAMETER(SAFETY=.9d0,PGROW=-.2d0,PSHRNK=-.25d0,ERRCON=1.89d-4)
  h=htry
  call rkck(y,dydx,n,x,h,ytemp,yerr,derivs)
  errmax=0.d0
  do 11 i=1,n
    errmax=max(errmax,abs(yerr(i)/yscal(i)))
11 continue
  errmax=errmax/eps
  2 if(errmax.gt.1.d0)then
    htemp=SAFETY*h*(errmax**PSHRNK)
    h=sign(max(abs(htemp),.1d0*abs(h)),h)
    xnew=x+h
    if(xnew.eq.x) then ! if new step is the same as the previous failed
                      ! step, force a step of hadv
      h = hadv
      errmax = 1.d0
      goto 2
    endif
    goto 1
  else
    if(errmax.gt.ERRCON)then
      hnext=SAFETY*h*(errmax**PGROW)
    else
      hnext=5.d0*h
    endif
    hdid=h
    x=x+h

```

```

do 12 i=1,n
  y(i)=ytemp(i)
  continue
return
endif
END

! the following routines are from Press et al. 1992

FUNCTION ran2(idum)
FUNCTION betacf(a,b,x)
FUNCTION beta(a,b)
FUNCTION error(x)
FUNCTION gammaln(xx)
FUNCTION gammp(a,x)
SUBROUTINE indexx(n,arr,indx)
SUBROUTINE gcf(gammcf,a,x,gln)
SUBROUTINE gser(gamser,a,x,gln)
SUBROUTINE spline(x,y,n,yp1,ypn,y2)
SUBROUTINE odeint(ystart,nvar,x1,x2,eps,h1,hadv,nok,nbad,derivs,
SUBROUTINE rkck(y,dydx,n,x,h,yout,yerr,derivs)

! the following three routines are exact duplicates of odeint,rkqs, and rkck
! they exist to avoid calling odeint when already inside an integral
SUBROUTINE odeint2(ystart,nvar,x1,x2,eps,h1,hadv,nok,nbad,derivs,
*rkqs2)

SUBROUTINE rkqs2(y,dydx,n,x,htry,hadv,eps,yscal,hdid,hnext,derivs)
SUBROUTINE rkck2(y,dydx,n,x,h,yout,yerr,derivs)

```

F Simulated GCS Evolution Code

Following is a code which when given a simulated initial GCS from the first code (see Appendix E and section 3.7) integrates each GC including destruction mechanisms that modify orbital parameters and cause mass-loss, to provide an evolved GCS.

```

PROGRAM timeev
IMPLICIT NONE
INTEGER dcmlpnts , objnum , modnum , KMAXX , NMAX , E2dtabn , l2dtabn
INTEGER mdfnun , kingdatn , csampn , shokn , mbarn , dummynum , rofpsin
REAL*8 PI
PARAMETER ( dcmlpnts=1 , objnum=100000 , modnum=1000 , KMAXX=100 , NMAX=50 )
PARAMETER ( E2dtabn=20 , l2dtabn=80 , rofpsin=200 )
PARAMETER ( mdfnun=2000 , PI=3.141592654d0 , kingdatn=193 , csampn=500 )
PARAMETER ( shokn=500 , mbarn=500 , dummynum=2000 )

REAL*8 kingSig0 ( kingdatn ) , kingc ( kingdatn ) , kingRc ( kingdatn )
REAL*8 kingrh ( kingdatn ) , kinglosrh ( kingdatn ) , kingl dsig0 ( kingdatn )
REAL*8 kinglossig0 ( kingdatn ) , kingM ( kingdatn ) , kingrtrh ( kingdatn )
REAL*8 kingKE ( kingdatn ) , kingBE ( kingdatn ) , spd2rtrhdc2 ( kingdatn )
REAL*8 kingW0 ( kingdatn ) , spd2W0dc2 ( kingdatn )
REAL*8 spd2MdW02 ( kingdatn ) , kingrhot ( kingdatn )
REAL*8 spd2rhotdW02 ( kingdatn )

REAL*8 dummy1 ( dummynum )

REAL*8 psi ( rofpsin ) , rofpsi ( rofpsin ) , spd2rdpsi2 ( rofpsin ) , rppsi
REAL*8 psi0 , psifunc , Ni , rapsi , rhohok , rhohmean
REAL*8 lconst , Emin , Emax , N1 , Nn , hpt , newsmall , newbig

REAL*8 lgml , lgmwd , lgmn , lgmu , Pwd , Pns , Pbh , spd2mbardt2 ( mbarn )
REAL*8 dlgtodlgtot , logmto ( mbarn ) , logtev ( mbarn ) , logmbar ( mbarn )
REAL*8 Mnorm , Nnorm , Mev , Nev , lgtpt , dlgmbar dlgt ( mbarn )
REAL*8 spd3mbardt3 ( mbarn )

REAL*8 cumulativec ( csampn ) , cvec ( csampn ) , d2cdf ltc2 ( csampn )

REAL*8 lgchi ( shokn ) , shokrp ( shokn ) , d2chidrp2 ( shokn )
REAL*8 rppt , I0 , I1 , J0 , J1

REAL*8 objrgci ( objnum ) , objphii ( objnum ) , objlgMi ( objnum )
REAL*8 objEi ( objnum ) , objhi ( objnum ) , objvri ( objnum )
REAL*8 objvthetai ( objnum ) , objrpi ( objnum ) , objrLi ( objnum )
REAL*8 objvphii ( objnum ) , objrEi ( objnum ) , objrai ( objnum )
REAL*8 objecci ( objnum ) , objvti ( objnum ) , objTri ( objnum )
REAL*8 objrhoji ( objnum ) , objrhoji ( objnum ) , objW0i ( objnum )
REAL*8 objrbari ( objnum ) , objLi ( objnum ) , objjbari ( objnum )

REAL*8 objLf ( objnum ) , objlgMf ( objnum ) , objvrf ( objnum )
REAL*8 objvphif ( objnum ) , objvthetaf ( objnum ) , objrhojf ( objnum )
REAL*8 objraf ( objnum ) , objEf ( objnum ) , objrgcf ( objnum )
REAL*8 objrpf ( objnum ) , objTrf ( objnum ) , objhf ( objnum )
REAL*8 objecf ( objnum ) , objrhojf ( objnum )

REAL*8 okobjLi ( objnum ) , okobjEi ( objnum ) , okobjrgci ( objnum )
REAL*8 okobjlgMi ( objnum ) , okobjrhoji ( objnum ) , okobjrhoji ( objnum )
REAL*8 okobjhi ( objnum ) , okobjvri ( objnum ) , okobjvphii ( objnum )
REAL*8 okobjvthetai ( objnum ) , okobjrpi ( objnum ) , okobjrai ( objnum )
REAL*8 okobjecci ( objnum ) , okobjTri ( objnum )

REAL*8 lgphi ( modnum ) , lgr ( modnum ) , dlghid lgr ( modnum ) , lgrho ( modnum )
REAL*8 d2lgphid lgr2 ( modnum ) , lgvc2 ( modnum ) , lgM ( modnum )
REAL*8 lgEc ( modnum ) , logLc ( modnum ) , spd2LcdEc2 ( modnum )
REAL*8 spd2phidr2 ( modnum )
REAL*8 spd3phidr3 ( modnum ) , spd4phidr4 ( modnum ) , spd2rhodr2 ( modnum )
REAL*8 spd2rdphi2 ( modnum ) , lgvd2 ( modnum ) , spd2vd2dr2 ( modnum )
REAL*8 lgomegac2 ( modnum ) , spd2omegac2dr2 ( modnum )

REAL*8 mdflgE ( mdfnun ) , mdflgEr ( mdfnun ) , mdflgisodf ( mdfnun )
REAL*8 mdflggdf ( mdfnun ) , mdfDED ( mdfnun ) , mdfnDED ( mdfnun )
REAL*8 mdflLc ( mdfnun ) , spd2fdE2 ( mdfnun )

INTEGER nvar , nok , nbad , kmax , kount , hount , ios
REAL*8 ystart ( NMAX ) , xi , xf , eps , h1 , hadv , dxsav , xp ( KMAXX )
REAL*8 yp ( NMAX , KMAXX )

REAL*8 dumml , dumm2 , dumm3 , dumm4 , dumm5 , dumm6 , dumm7 , dumm8

```

```

INTEGER strind1, strind2, strind3, strlen, modeltype, raninit
INTEGER evaptype, shoktype, frictype, rtidtype, clevtype
INTEGER rho0type, stevtype, i, j, n, splinterrn, conctype, iok, reqnum
REAL*8 rho0param1, rho0param2, cmin, Lvec2d(3*E2dtabn, 3*l2dtabn), tf
REAL*8 logE, lgrbar, lgrEpt, lgjbartab(3*E2dtabn, 3*l2dtabn), lgjbar, k
REAL*8 E2dtab(3*E2dtabn), l2dtab(3*l2dtabn), cmax, fltcpt, cpt, rtrhpt
REAL*8 lgrptab(3*E2dtabn, 3*l2dtabn), gamma, hfunc, lgrppt, lgrapt, p0
REAL*8 lgratab(3*E2dtabn, 3*l2dtabn), yp1, ypn, bisacc, rmaxpt
REAL*8 spd2rcdEc2(modnum), spd2rcdLc2(modnum), rEfind, LcofE
REAL*8 rtbis, lgTrtab(3*E2dtabn, 3*l2dtabn), lgEpt, m0overM0, lgtmin
REAL*8 logLpt, ran2, rhojev, lsafety, Esafety
REAL*8 lgrbartab(3*E2dtabn, 3*l2dtabn), lpt, rhoh0, W0
REAL*8 phiofr, time1, time2, smalllofL
REAL*8 ei, Lminvec(3*E2dtabn)
REAL*8 ELminvec(3*E2dtabn), spd2Lminde2(3*E2dtabn), BigLofl, Esup
REAL*8 rminpt, rofphi, rref, rhoj0, alpha, bisE, bisL
REAL*8 rhojbar, G, rmin, rhojpower, W0pt, mbar, Mi, endtime, lgTr
REAL*8 starttime, mtooft, opfn, ChabIMF, Pej, mrm, Mofr, dimlmin, dimlmax
REAL*8 Mgaltot, logL, lgra, lgrp, lgrE, lgrL, Lmin, mto, Lmax, rmax, Einf
LOGICAL linin, nzd, valid, kcr, initint, lok, Eok, abort
CHARACTER*15 strnum, bis
CHARACTER*30 objfilename, modfilename, mdfilename, splintcall
CHARACTER*30 dfrfilename, evlfilename
CHARACTER*650 line
COMMON /abortpath/ tf, abort
COMMON /splintpath/ splintcall
COMMON /Lcpath/ spd2LcdEc2, logLc, lgEc
COMMON /Lminpath/ Lminvec, ELminvec, spd2Lminde2
COMMON /lgrpath/ lgr
COMMON /chipath/ lgchi, shokrp, d2chidrp2
COMMON /pmfixpath/ rppt, Mgaltot
COMMON /rhopath/ lgrho, spd2rhodr2
COMMON /omegacpath/ lomegac2, spd2omegac2dr2
COMMON /kingpath/ kingW0, kingM, kingrhot, spd2MdW02, spd2rhotdW02
COMMON /Ecpath/ spd2rcdEc2, spd2rcdLc2
COMMON /isodfpath/ mdflgE, mdflgisodf, spd2fdE2
COMMON /phiandr/ lgphi, dlghidlgr, d2lgphidlgr2, spd2phidr2,
*          spd3phidr3, spd4phidr4, spd2rdphi2
COMMON /vd2path/ lgvd2, spd2vd2dr2
COMMON /orbparampath/ lgrp, lgra, lgrbar, lgjbar, lgTr
COMMON /rappath/ lgrppt, lgrapt
COMMON /paraminterp/ lgrptab, lgratab, lgrbartab, lgjbartab, lgTrtab,
*          E2dtab, l2dtab
COMMON /parampath/ p0, G, m0overM0
COMMON /alphapath/ alpha
COMMON /clusterparams/ mbar, Mi, rhoj0, rhoh0, W0, gamma, ei
COMMON /deadpath/ rmin, rmax, Lmin, Lmax, Esup, Einf
COMMON /ELpath/ logE, hpt
COMMON /azpath/ rofpsi, psi, spd2rdpsi2
COMMON /hfuncpath/ lgrEpt, rmaxpt
COMMON /rhojint/ rhojpower
COMMON /rjbarpath/ rhojbar
COMMON /evappath/ evaptype
COMMON /shokpath/ shoktype
COMMON /stevpath/ stevtype
COMMON /rtidpath/ rtidtype
COMMON /fricpath/ frictype
COMMON /clevpath/ clevtype
COMMON /rho0path/ rho0param1, rho0param2, conctype
COMMON /modelpath/ k, rref, modeltype
COMMON /errpath/ splinterrn
COMMON /linpath/ linin
COMMON /SNSMpath/ initint
COMMON /mbarpath/ logtev, logmbar, dlgmbarldgt, spd2mbardt2,
*          spd3mbardt3
COMMON /stevparam/ lgml, lgmu, Pwd, Pns, Pbh, lgmwd, lgmns, Mnorm
COMMON /path/ kmax, kount, dxsav, xp, yp, hount
COMMON /bispath/ bisE, bisL, bis
EXTERNAL cint, rkqs, hfunc, rofpsiint, azint, psifunc, rbarint
EXTERNAL rjbarfunc, I0int, I1int, J0int, J1int, SMint, SNint, orbint
EXTERNAL jbarint

yp1 = 1.d30
ypn = 1.d30
p0 = 9.d0
alpha = 0.d0
rhojpower = 5.d-1
bisacc = 1.d-40
initint = .false. ! shortcut for calculating stellar IMF normalisations
linin = .false. ! setting this to true will force all interpolation to be linear,
! setting false allows linintrp argument in splintcheck to chose.
nzd = .false.

```



```

kcr = .false.
! age of system in Myr at which output from first code is considered
! 'initial' - post supernovae and 'infant mortality' phase
lgtmin = log10(30.d0)
! gravitational constant in units of pc^3 Gyr^-2 M_0^-1,
! only used for C in evaporation routine
G = 4499.753331d0

write(6,1000) 'input_desired_model_type:'
write(6,1000) '1:_Polytropes'
write(6,1000) '2:_King'
write(6,1000) '3:_Dehnen'
write(6,1000) '4:_Powerlaws'
read(5,*) modeltype
write(6,1000) 'input_desired_model_defining_parameter'
if (modeltype.eq.1) then
  write(6,1000) 'input_k:'
  objfilename = 'polytrope_objct_k_'
  modfilename = 'polytrope_model_k_'
  mdffilename = 'polytrope_distg_k_'
  dfrfilename = 'polytrope_dynfr_k_'
  evlfilename = 'polytrope_evolv_k_'
else if (modeltype.eq.2) then
  write(6,1000) 'input_W0:'
  objfilename = 'king_objct_W0_'
  modfilename = 'king_model_W0_'
  mdffilename = 'king_distg_W0_'
  dfrfilename = 'king_dynfr_W0_'
  evlfilename = 'king_evolv_W0_'
else if (modeltype.eq.3) then
  write(6,1000) 'input_gamma:'
  objfilename = 'dehnen_objct_G_'
  modfilename = 'dehnen_model_G_'
  mdffilename = 'dehnen_distg_G_'
  dfrfilename = 'dehnen_dynfr_G_'
  evlfilename = 'dehnen_evolv_G_'
else if (modeltype.eq.4) then
  write(6,1000) 'input_k:'
  objfilename = 'powerlaw_objct_k_'
  modfilename = 'powerlaw_model_k_'
  mdffilename = 'powerlaw_distg_k_'
  dfrfilename = 'powerlaw_dynfr_k_'
  evlfilename = 'powerlaw_evolv_k_'
end if
read(5,*) k

write(6,1000) 'input_desired_evolved_cluster_system_age_in_Myr'
read(5,*) tf

write(6,1001) 'input_desired_stellar_mass_to_galaxy_scale_mass',
*                                     '_ratio:',
read(5,*) m0overM0

WRITE (strnum,*) (k+5.d0*(1.d1**(-1.d0-1.d0*(dcmlpnts))))

strind1 = index(strnum, '.')
strnum = strnum(1:strind1+dcmlpnts)

strlen = len(strnum) ! file naming stuff
strind1 = index(objfilename, '_') ! find position of first blank space in
! savfilename
strind2 = index(strnum, '.') ! find position of decimal place in W0 string
1 strind3 = index(strnum, '_') ! find position of first blank space in W0
! string
if (strind3.eq.1) then ! remove all leading blank spaces
  strnum=strnum(strind3+1:strlen)
  goto 1
end if ! loops until first character of strnum is not a blank space
strind2 = index(strnum, '.') ! find new position of decimal palce
strind3 = index(strnum, '_') ! find new position of first blank space, now
! somewhere at end of string
if (strind3.eq.strind2+1) then ! if there are no zeros after the decimal
! place, add as many as are required
  do i = 1,dcmlpnts
    strnum = strnum(1:(strind2+i-1)) // '0'
  end do
  strind3 = index(strnum, '_') ! find position of first blank space again
end if
objfilename=objfilename(1:strind1-1)//strnum(1:strind3-1)//'.out'
modfilename=modfilename(1:strind1-1)//strnum(1:strind3-1)//'.out'
mdffilename=mdffilename(1:strind1-1)//strnum(1:strind3-1)//'.out'
dfrfilename=dfrfilename(1:strind1-1)//strnum(1:strind3-1)//'.out'
evlfilename=evlfilename(1:strind1-1)//strnum(1:strind3-1)//'.out'

```

```

open(4,file=modfilename,status="old")
i = 0
do j=1,2147483646
  read(4,1000,end=100) line
  if(line(1:1).ne. '#') then
    i = i + 1
    read(line,*) lgphi(i),lgr(i),dlgphidlgr(i),lgrho(i),
*      d2lgphidlgr2(i),lgvc2(i),lgM(i),lgvd2(i),lgEc(i)
*      ,logLc(i)
    end if
  end do
100 close(4)

rmin = lgr(1)
rmax = lgr(modnum)
Einf = lgEc(1)
Esup = lgphi(modnum)
Lmin = 0.5*log10(2.d0*(1.d1*(Esup-Einf)))+rmin
Lmax = logLc(modnum)

if (modeltype==1.or.modeltype==2) then
  rref=0.d0
else
  rref=1.d1*(rmin-1.d0)
end if

do i = 1,modnum
  lgomegac2(i) = lgvc2(i) - 2.d0*lgr(i)
end do

open(4,file=mdffilename,status="old")
i = 0
do j=1,2147483646
  read(4,1000,end=101) line
  if(line(1:1).ne. '#') then
    i = i + 1
    read(line,*) mdflgE(i),mdflgEr(i),mdflgisodf(i),mdflggdf(i),
*      mdfDED(i),mdfnDED(i),mdfLc(i)
  end if
end do
101 close(4)

open(4,file=objfilename,status="old")
i = 0
do j=1,2147483646
  read(4,1000,end=102) line
  if(line(1:1).ne. '#') then
    i = i + 1
    read(line,*) objrgci(i),objphii(i),objlgMi(i),objEi(i),
*      objhi(i),objvri(i),objvphii(i),objvthetai(i),
*      objrpi(i),objrLi(i),objrEi(i),objrai(i),
*      objecci(i)
  end if
end do
102 close(4)

do i = 1,objnum
  objvti(i) = ((objvphii(i)**2.d0)+(objvthetai(i)**2.d0))*5.d-1
  objLi(i) = objrgci(i)*objvti(i)
end do

CALL spline(lgr,lgrho,modnum,yp1,ypn,spd2rhodr2)
CALL spline(lgr,lgomegac2,modnum,yp1,ypn,spd2omegac2dr2)
CALL spline(lgr,lgphi,modnum,yp1,ypn,spd2phidr2)
CALL spline(lgphi,lgr,modnum,yp1,ypn,spd2rdphi2)
CALL spline(lgr,dlgphidlgr,modnum,yp1,ypn,spd3phidr3)
CALL spline(lgr,d2lgphidlgr2,modnum,yp1,ypn,spd4phidr4)
CALL spline(lgEc,logLc,modnum,yp1,ypn,spd2LcdEc2)
CALL spline(lgEc,lgr,modnum,yp1,ypn,spd2rcdEc2)
CALL spline(logLc,lgr,modnum,yp1,ypn,spd2rcdLc2)
CALL spline(lgr,lgvd2,modnum,yp1,ypn,spd2vd2dr2)
CALL spline(mdflgE,mdflgisodf,mdfnum,yp1,ypn,spd2fdE2)

4 write(6,1000) 'input_desired_evaporation_prescription:'
write(6,1000) '0:_none'
write(6,1000) '1:_King_1966'
write(6,1000) '2:_Vesperini_&_Heggie_1997'
write(6,1000) '3:_Fall_&_Zhang_2001'
write(6,1000) '4:_Jordan_2007'
write(6,1000) '5:_Chapter_2_Virgo_result'
write(6,1000) '6:_McLaughlin_&_Fall_2008'
write(6,1000) '7:_Chandar_et_al._2007'

```

```

write(6,1000) '8:_Baumgardt_1998'
write(6,1000) '9:_Gieles_&_Baumgardt_2008'
write(6,1000) '10:_Gieles_et_al._2011'
write(6,1000) '11:_Gieles_&_Baumgardt_2008_non-linear'
write(6,1000) '12:_Gieles_et_al._2011_non-linear'
write(6,1000) '13:_Tanikawa_&_Fukushige_2010'
write(6,1000) '14:_Lamers_et_al._2010'
write(6,1000) '15:_Takahashi_&_Baumgardt_2012'
read(5,*) evaptype
valid = .false.
do i = 0,15
  if (evaptype.eq.i) valid = .true.
end do
if (valid.eqv..false.) then
  write(6,1000) 'invalid_selection'
  goto 4
end if
! single out prescriptions that require non-zero initial half-mass density
if (evaptype.eq.4) nzd = .true.
if (evaptype.eq.5) nzd = .true.
if (evaptype.eq.6) nzd = .true.
if (evaptype.eq.7) nzd = .true.
if (evaptype.eq.9) nzd = .true.
if (evaptype.eq.11) nzd = .true.
if (evaptype.eq.13) nzd = .true.
if (evaptype.eq.14) nzd = .true.
if (evaptype.eq.15) nzd = .true.
! single out prescriptions that require clusters
! with a W0 (e.g. king 1966 model clusters)
if (evaptype.eq.1.or.evaptype.eq.14) then
  kcr = .true.
end if

5  write(6,1000) 'input_desired_shock_prescription:'
write(6,1000) '0:_none'
write(6,1000) '1:_Gnedin,_Hernquist_&_Ostriker_1999'
read(5,*) shoktype
valid = .false.
do i = 0,1
  if (shoktype.eq.i) valid = .true.
end do
if (valid.eqv..false.) then
  write(6,1000) 'invalid_selection'
  goto 5
end if
if (shoktype.eq.1) then
  nzd = .true.
end if

6  write(6,1000) 'input_desired_stellar_evolution_prescription:'
write(6,1000) '0:_none'
write(6,1000) '1:_Chernoff_&_Weinberg_1990'
write(6,1000) '2:_Lamers_et_al._2010'
read(5,*) stevtype
valid = .false.
do i = 0,2
  if (stevtype.eq.i) valid = .true.
end do
if (valid.eqv..false.) then
  write(6,1000) 'invalid_selection'
  goto 6
end if

do i = 1,mbarn ! need this whether stellar evolution is on or off
  logtev(i) = lgtmin+(i-1)*(log10(tf)-lgtmin)/(1.d0*(mbarn-1))
end do
if (stevtype.eq.1) then ! author specific stuff
  lgmwd = log10(4.7d0)
  ! this is actually more like lgmwd, but Chernoff & Weinberg have no
  ! remnant between 4.7 and 8.0 solar, and dont include black holes at all
  lgmns = log10(8.d0)
else if (stevtype.eq.2) then ! Hurely et al. 2000
  lgmwd = log10(8.19d0)
  lgmns = log10(24.76d0)
end if
if (stevtype.ne.0) then ! generic integrals and definitions
  do i = 1,mbarn ! define time and turnoff mass arrays
    logmto(i) = log10(mtooft(logtev(i)))
  end do
13 if (stevtype.eq.1) then
    print *, 'WD_if_log_m<', lgmwd
    print *, 'no_remnant_if', lgmwd, '<=<log_m<=<', lgmns
    print *, 'NS_if_log_m>', lgmns

```

```

      write (6,1001) 'input_desired_ejection_probability_for_',
*      'remnant_WD_and_NS_upon_formation'
      ! Pns is used to remove massless remnants, Pbh is actually
      ! treating ejection probability of NSs
      read(5,*) Pwd,Pbh
      ! massless remnant will still be counted if
      ! it is not treated as being ejected
      Pns = 1.d0
      else if (stevtype.eq.2) then
      print *, 'WD_if_log_m<',lgmwd
      print *, 'NS_if',lgmwd,'<=log_m<=',lgmns
      print *, 'BH_if_log_m>',lgmns
      write (6,1001) 'input_desired_ejection_probability_for_',
*      'remnant_WD, NS, and BH_upon_formation'
      read(5,*) Pwd,Pns,Pbh
      end if
      if (Pwd.lt.0.d0.or.Pwd.gt.1.d0.or.Pns.lt.0.d0.or.Pns.gt.1.d0
*      .or.Pbh.lt.0.d0.or.Pbh.gt.1.d0) then
      print *, 'all_probabilities_must_satisfy_0<=P<=1'
      goto 13
      end if
14 write (6,1001) 'input_desired_lower_and_upper_log_stellar_',
*      'mass_limits, lgml_and_lgmu'
      print*, 'lgmu_must_be_less_than_or_equal_to_',
*      log10(mtooft(lgtmin))
      read(5,*) lgml,lgmu
      if (lgml.ge.lgmu) then
      print *, 'lower_mass_must_be_less_than_upper_mass'
      goto 14
      end if

      nvar = 1
      kmax = 0
      eps = 1.d-7
      hl = 1.d-10
      hadv = 1.d-10
      initint = .true.
      xi = lgml
      xf = lgmu
      ystart(1) = 0.d0
      CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,SNint,rkqs)
      Nnorm = ystart(1)
      ystart(1) = 0.d0
      CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,SMint,rkqs)
      Mnorm = ystart(1)
      initint = .false.
      do i = 1,mbarn
      lgtpt = logtev(i)
      xi = logmto(i)
      xf = lgmu
      ystart(1) = 0.d0
      if (xi.lt.xf) then ! if mto is greater than mu, then no stars are expiring yet
      CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,SNint
*      ,rkqs)
      end if
      Nev = 1.d0 - ystart(1)/Nnorm
      ystart(1) = 0.d0
      if (xi.lt.xf) then ! if mto is greater than mu, then no stars are expiring yet
      CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,SMint
*      ,rkqs)
      end if
      Mev = 1.d0 - ystart(1)/Mnorm
      logmbar(i) = log10(Mev/Nev)
      mto = 1.d1*logmto(i)
      dlgbardlgt(i) = ChabIMF(logmto(i))*dlgmtodlgtft(logtev)*
*      ((mto-(1.d0-Pej(logmto(i)))*mrm(logmto(i)))/
*      Mev-Pej(logmto(i))/Nev)/log(1.d1)
      end do
      CALL spline(logtev,logmbar,mbarn,yp1,ypn,spd2mbardt2)
      write(45,1010) '#', 'log_t', 'log_mbar', 'log_mto',
*      'd_log_mto/d_log_t', 'd_N/d_log_m', 'Pej(mto)', 'log_mrm(mto)',
*      'd_log_mbar/d_log_t'
      do i = 1,mbarn
      write(45,1008) logtev(i),logmbar(i),logmto(i)
*      ,dlgmtodlgtft(logtev),ChabIMF(logmto(i)),Pej(logmto(i))
*      ,log10(mrm(logmto(i))),dlgbardlgt(i)
      end do
      else ! if stellar evolution is turned off
      do i = 1,mbarn
      logmbar(i) = 0.d0 ! mbar is constant at all times
      dlgbardlgt(i) = 0.d0
      spd2mbardt2(i) = 0.d0
      end do

```

```

end if

7   write(6,1000) 'input_desired_dynamical_friction_prescription:'
   write(6,1000) '0:_none'
   write(6,1000) '1:_Chandrasekhar_1943'
   read(5,*) frictype
   valid = .false.
   do i = 0,1
     if (frictype.eq.i) valid = .true.
   end do
   if (valid.eqv..false.) then
     write(6,1000) 'invalid_selection'
     goto 7
   end if

8   write(6,1000) 'input_desired_tidal_radius_definition:'
   write(6,1000) '1:_orbital_pericentre'
   write(6,1000) '2:_time_averaged_galactocentric_radius'
   write(6,1001) '3:_rgc_where_instantaneous_sqrt_jacobi_density',
*               'equals_time_average_of_sqrt_jacobi_density'
   write(6,1001) '4:_rgc_where_jacobi_density_equals_jacobi_density'
*               'at_pericentre_times_function_of_ellipticity'
   read(5,*) rtdtype
   valid = .false.
   do i = 1,4
     if (rtdtype.eq.i) valid = .true.
   end do
   if (valid.eqv..false.) then
     write(6,1000) 'invalid_selection'
     goto 8
   end if

9   write(6,1001) 'input_desired_initial_cluster_half-mass_density',
*               '_method:'
   write(6,1001) '0:_none_(all_clusters_will_be_considered_to_be',
*               'point_masses)'
   write(6,1001) '1:_randomly_assigned_king_concentration_plus',
*               '_tidal_density_at_time_averaged_radius'
   write(6,1000) '2:_half-mass_radius/jacobi_radius~mass_relation'
   write(6,1000) '3:_half-mass_radius~mass_relation'
   read(5,*) rho0type
   if (rho0type.eq.0.and.nzd) then
     write(6,1001) 'selected_prescriptions_require',
*               '_non-zero_initial_half-mass_density'
     goto 9
   end if
   if (rho0type.ne.1.and.kcr) then
     write(6,1001) 'selected_prescriptions_require_king_model',
*               '_clusters_(option_1)'
     goto 9
   end if
   if (rho0type.eq.1) then
     open(4,file='mount_nfs/users/kmt/kingmod.dat',status='old')
     i = 0
     do j=1,2147483646
       read(4,1000,end=103) line
       if (line(1:1).ne.'#') then
         i = i + 1
         read(line,*) kingW0(i),kingc(i),kingRc(i),kingrh(i),
*               kinglosrh(i),kingldsig0(i),kinglossig0(i),
*               kingM(i),kingSig0(i),kingKE(i),kingBE(i)
       end if
     end do
     close(4)
     do i = 1,kingdatn
       kingW0(i) = log10(KingW0(i))
       kingc(i) = log10(kingc(i))
       kingrtrh(i) = kingc(i)-log10(kingrh(i))
       kingM(i) = log10(kingM(i))
       kingrhot(i) = kingM(i)-3.d0*kingc(i)-log10(4.d0*PI/3.d0)
     end do
     CALL spline(kingc,kingrtrh,kingdatn,yp1,ypn,spd2rtrhdc2)
     CALL spline(kingc,kingW0,kingdatn,yp1,ypn,spd2W0dc2)
     CALL spline(kingW0,kingM,kingdatn,yp1,ypn,spd2MdW02)
     CALL spline(kingW0,kingrhot,kingdatn,yp1,ypn,spd2rhotdW02)
103  write(6,1001) 'enter_negative_integer_for_random_number',
*               '_generator_seed'
     read(5,*) raninit
     if (raninit.ge.0) goto 10
     write(6,1001) 'input_desired_initial_concentration',
*               '_distribution:'
     write(6,1000) '1:_Gaussian'
     write(6,1000) '2:_Schechter'

```

```

11      read(5,*) conctype
      write(6,1004) 'input_desired_minimum_and_maximum_allowed ',
*      'concentrations:_(minimum=,kingc(1),_maximum=,
*      kingc(kingdatn),)'
      read(5,*) cmin,cmax
      if (cmin.lt.kingc(1)) then
        write(6,1000) 'minimum_concentration_too_small'
        goto 11
      end if
      if (cmax.gt.kingc(kingdatn)) then
        write(6,1000) 'maximum_concentration_too_big'
        goto 10
      end if
      if (conctype.eq.1) then
        write(6,1000) 'input_Gaussian_mean_and_variance'
        read(5,*) rho0param1,rho0param2
      else if (conctype.eq.2) then
        write(6,1001) 'input_power_law_slope_and_exponential_cut_off'
*      ,'_value'
      read(5,*) rho0param1,rho0param2
      end if
    else if (rho0type.eq.2) then
      write(6,1000) 'input_alpha_and_beta'
      read(5,*) rho0param1,rho0param2
    else if (rho0type.eq.3) then
      write(6,1000) 'input_alpha_and_beta'
      read(5,*) rho0param1,rho0param2
    end if
    write(6,1001) 'input_desired_internal_cluster_evolution ',
*    '_prescription:'
    write(6,1000) '0:_none'
    write(6,1000) '1:_Virial_evolution'
    read(5,*) clevtype

    if (shoktype.eq.1) then
      nvar = 1
      kmax = 0
      eps = 1.d-7
      hl = 1.d-10
      xi = 0.d0
      xf = 1.d40
      hadv = (xf-xi)*1.d-6
      Mgalatot = Mofr(1.d1**lgr(modnum)) ! bit unsure about this
      print *, 'calculating_interpolation_tables_for_distended_mass'
*      ,'_shocking_correction'
      CALL cpu_time(time1)
      do i = 1,shokn
        shokrp(i) =lgr(1)+(i-1)*(lgr(modnum)-lgr(1))/(1.d0*(shokn-1))
        rppt = 1.d1**shokrp(i)
        ystart(1) = 0.d0
        CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,I0int
*      ,rkqs)
        I0 = ystart(1)
        ystart(1) = 0.d0
        CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,I1int
*      ,rkqs)
        I1 = ystart(1)
        ystart(1) = 0.d0
        CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,J0int
*      ,rkqs)
        J0 = ystart(1)
        ystart(1) = 0.d0
        CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,J1int
*      ,rkqs)
        J1 = ystart(1)
        lgchi(i) = log10(5.d-1*((3.d0*J0-J1-I0)**2.d0)+
*      ((2.d0*I0-I1-3.d0*J0+J1)**2.d0)+(I0**2.d0)))
        print *, i,shokrp(i),I0,I1,J0,J1,lgchi(i)
      end do
      CALL spline(shokrp,lgchi,shokn,yp1,ypn,d2chidrp2)
      CALL cpu_time(time2)
      print *, 'shock_interpolation_tables_done.'
      print *, 'Time_taken= ',time2-time1
    end if

    OPEN(4,FILE=dfrfilename,STATUS="OLD",IOSTAT=ios,ERR=746)
    i = 1
    j = 1
    do n=1,2147483646
      if (n==1) print *, 'reading_orbital_parameter_tables_from_',
*      dfrfilename
      read(4,1000,end=747) line
      if(line(1:1).ne.'#') then

```

```

      read(line,*) dumm1,dumm2,dumm3,dumm4,dumm5,dumm6,dumm7,dumm8
      if (j==3*l2dtabn+1) then
        i = i + 1
        j = 1
      end if
      E2dtab(i) = dumm1
      l2dtab(j) = dumm2
      Lvec2d(i,j) = dumm3
      lgTrtab(i,j) = dumm4
      lgrptab(i,j) = dumm5
      lgrbartab(i,j) = dumm6
      lgjbartab(i,j) = dumm7
      lgratab(i,j) = dumm8
      j = j + 1
    end if
  end do
747  CLOSE(4)

  do i = 1,3*E2dtabn
    Lminvec(i) = rmin + 5.d-1*log10(1.d1**E2dtab(i)-phiofr(rmin))
    *      + 5.d-1*log10(2.d0)
    ELminvec(i) = E2dtab(i)
  end do
  CALL spline(ELminvec,Lminvec,3*E2dtabn,yp1,ypn,spd2LmindE2)

746  if (ios.ne.0) then ! interpolation tables for pericentre, apocentre,
                        ! time averaged radii, and whatever else.
    Emin = 1.d1*lgEc(1)+1.d-8
    Emax = 1.d1*lgphi(modnum)
    N1 = 1.d-3 ! controls how dense the point distribution around Emax is - smaller
                ! number = more dense
    Nn = Emax+N1-Emin

    print *, 'calculating_orbital_parameters_as_functions_of_E_and'
    *, '_L...'
    CALL cpu_time(time1)
    do i = 1,E2dtabn
      ! log of linear spacing with even point distribution
      E2dtab(i) = log10(Emin+i*(Emax-Emin)/(1.d0*(E2dtabn+1)))
      ! log of log spacing with large cluster of points at low end
      E2dtab(i+E2dtabn) = log10(Emin)+(i)*log10(Emax/Emin)/
                          (1.d0*E2dtabn+1)
      ! arbitrary convolution array
      Ni = log10(N1) + (i-1)*log10(Nn/N1)/(1.d0*(E2dtabn-1))
      ! log of log spacing arranged to have large cluster of points at high end
      E2dtab(i+2*E2dtabn) = log10(Emax+N1-(1.d1**Ni))
    end do
    CALL multisort(1,3*E2dtabn,E2dtab,dummy1,dummy1,dummy1,dummy1,
    *      dummy1,dummy1,dummy1)

    dimlmin = 1.d-4 ! smallest l allowed - depending on E this could be close
                    ! to either circular or radial orbit
    dimlmax = 0.9999d0 ! maximum allowed dimensionless angular momentum, just
                       ! short of a circular orbit
    N1 = 1.d-6 ! controls how dense the point distribution around dimlmax
                ! is - smaller number = more dense
    Nn = dimlmax+N1-dimlmin

    do i = 1,3*E2dtabn
      Lminvec(i) = rmin + 5.d-1*log10(1.d1**E2dtab(i)-phiofr(rmin))
      *      + 5.d-1*log10(2.d0)
      ELminvec(i) = E2dtab(i)
    end do
    CALL spline(ELminvec,Lminvec,3*E2dtabn,yp1,ypn,spd2LmindE2)

    do i = 1,l2dtabn
      ! arbitrary convolution array
      Ni = log10(N1) + (i-1)*log10(Nn/N1)/(1.d0*(l2dtabn-1))
      ! log of log spacing arranged to have large cluster of points at high end
      l2dtab(i) = log10(dimlmax+N1-(1.d1**Ni))
      ! log of log spacing to have large cluster of points at low end
      l2dtab(i+l2dtabn) = log10(dimlmin)+(i)*log10(dimlmax/dimlmin)
      *      / (1.d0*l2dtabn+1)
      ! log of linear spacing with even point distribution
      l2dtab(i+2*l2dtabn) = log10(dimlmin+(i-1)*(dimlmax-dimlmin))/
      *      (1.d0*l2dtabn)
    end do
    CALL multisort(1,3*l2dtabn,l2dtab,dummy1,dummy1,dummy1,dummy1,
    *      dummy1,dummy1,dummy1)

    psi0 = 1.d0
    lconst = 1.d0
    do i = 1,rofpsin

```



```

WRITE(7,1001) '#column_2:_l=[L-Lmin]/[Lc(E)-Lmin]'
WRITE(7,1000) '#column_3:_orbital_angular_momentum'
WRITE(7,1000) '#column_4:_radial_orbital_period'
WRITE(7,1000) '#column_5:_orbital_pericentre'
WRITE(7,1000) '#column_6:_time_averaged_orbital_radius'
WRITE(7,1001) '#column_7:_radius_where_jacobi_density_equals_',
*      'time_average_of_some_power_of_jacobi_density'
* WRITE(7,1000) '#column_8:_orbital_apocentre'
do i = 1,3*E2dtabn
  do j = 1,3*I2dtabn
    write(7,1003) E2dtab(i),I2dtab(j),Lvec2d(i,j),lgTrtab(i,j),
*      lgrptab(i,j),lgrbartab(i,j),lgjbartab(i,j),
*      lgratab(i,j)
  end do
end do
CLOSE(7)
CALL cpu_time(time2)
print *, 'orbital_parameters_as_functions_of_E_and_L_done.'
print *, 'Time_taken_=',time2-time1
end if

print *, 'allocating_rbar,_jbar,_Tr,_and_rhoj'
j = frictype
frictype = 1
do i = 1,objnum ! define initial time averaged radii, radii where sqrt jacobi
! density equals time average sqrt jacobi density, and radial
! period for all tracers
  lgEpt = log10(objEi(i))
  logLpt = log10(objLi(i))
  lpt = logLpt-log10(LcofE(lgEpt))
  objrbari(i) = opfn(lgEpt,logLpt,2)! 2d interpolation on E and L for rbar
  objjbari(i) = opfn(lgEpt,logLpt,3)! 2d interpolation on E and L for jbar
  objTri(i) = opfn(lgEpt,logLpt,5)! 2d interpolation on E and L for radial period
  objrhoji(i) = log10(rhojev(objrgci(i),lgEpt,logLpt))
end do
frictype = j

! initial density allocation
if (rho0type.eq.0) then
  do i = 1,objnum
    objrhoji(i) = 4.d1
  end do
else if (rho0type.eq.1) then ! assign initial king model to clusters based on randomly
! sampled user specified initial concentration function
  nvar = 1
  kmax = 0
  eps = 1.d-6
  h1 = 1.d-10
  do i = 1,csampn ! find cumulative distribution for specified concentration distribution
    xi = cmin
    xf = cmin+(i-1)*(cmax-cmin)/(1.d0*(csampn-1))
    hadv = 1.d-6*(xf-xi)
    ystart(1) = 0.d0
    CALL odeint(ystart,nvar,xi,xf,eps,h1,hadv,nok,nbad,cint,rkqs)
    cumulativec(i) = ystart(1)
    cvec(i) = xf
  end do
  do i = 1,csampn
    ! normalise cumulative distribution to run between 0 and 1
    cumulativec(i) = cumulativec(i)/cumulativec(csampn)
  end do
  CALL spline(cumulativec,cvec,csampn,yp1,ypn,d2cdfc2)
  rhohok = 0.d0
  rhohmean = 0.d0
  do i = 1,objnum
    fltcpt = ran2(raninit) ! draw a random number between 0 and 1
    splintcall = 'cvec_c_sampling'
    ! find value of concentration corresponding to cumulative
    ! distribution = random number
    CALL splintcheck(cumulativec,cvec,d2cdfc2,csampn,fltcpt,cpt
*      ,.false.)
*
    splintcall = 'king_W0_interpolation'
    ! find value of central potential corresponding to concentration
    CALL splintcheck(kingc,kingW0,spd2W0dc2,kingdatn,cpt,W0pt
*      ,.false.)
*
    splintcall = 'kingc_rtrh_interpolating'
    ! find value of rt/rh corresponding to concentration
    CALL splintcheck(kingc,kingrtrh,spd2rtrhdc2,kingdatn,cpt,
*      rtrhpt,.false.)
*
    ! allocation of concentration does not work for tidally
    ! compressed clusters, as rt=inf
    if (1.d0/objrhoji(i).eq.0.d0) then
      objrhoji(i) = rhohmean
    end if
  end do
end do

```

```

else
  objrhohi(i) = log10(5.d-1) + objrhoji(i) + 3.d0*rtrhpt
  rhohmean = (rhohmean*rhohok + objrhohi(i))/(rhohok+1.d0)
  rhohok = rhohok + 1.d0
end if
objW0i(i) = 1.d1**W0pt
end do
! assign initial ratio rt/rh to cluster based on user defined function
else if (rho0type.eq.2) then
  do i = 1,objnum
    rtrhpt = log10(rho0param1)+rho0param2*objlgMi(i)
    objrhohi(i) = log10(5.d-1) + objrhoji(i) + 3.d0*rtrhpt
  end do
! assign rh based on cluster M according to user defined function
else if (rho0type.eq.3) then
  do i = 1,objnum
    objrhohi(i) = log10(3.d0*m0overM0)-log10(2.d0*P0)-3.d0*
*      log10(rho0param1)+(1.d0-3.d0*rho0param2)*objlgMi(i)
    if (objrhohi(i).ne.objrhohi(i)) then
      print *, objrhohi(i)
      pause
    end if
  end do
end if
end if

print *, 'integrating_object_orbits'
nvar = 5
kmax = 0
eps = 1.d-5
hl = 1.d-10
hadv = 1.d-6 ! this is how much the integration variable
               ! is pushed along by if the integration gets stuck
xi = lgtmin
xf = log10(tf)
lsafety = 1.d-3 ! absolute
Esafety = log10(5.d-2) ! relative
iok = 1
reqnum = objnum
CALL CPU.TIME(starttime)
do i = 1,objnum
  lgEpt = log10(objphii(i)+5.d-1*((objvri(i)**2.d0)+(objvti(i)**
*      2.d0)))
  logLpt = log10(objrgci(i))+log10(objvti(i))
  lpt = smalllofL(logLpt,lgEpt)
  if (Esup-1*lgEpt.gt.Esafety.and.lgEpt-Einf.gt.Esafety)Eok=.true.
  if (lpt.gt.lsafety.and.lpt.lt.1.d0-lsafety) lok=.true.
  if (Eok.and.lok.and.iok.le.reqnum) then
    ! logarithmic cluster position
    ystart(1) = log10(objrgci(i))
    ! logarithmic cluster position time derivative
    ystart(2) = 1.d1*(xi-log10(objrgci(i)))*objvri(i)
    ! logarithmic cluster tangential speed
    ystart(3) = log10(objvti(i))
    ! logarithmic cluster mass
    ystart(4) = objlgMi(i)
    ! logarithmic cluster half-mass density
    ystart(5) = objrhohi(i)
    lgrp = objrpi(i)
    lgra = objrai(i)
    lgrbar = objrbari(i)
    lgjbar = objjbbari(i)
    lgrE = objrEi(i)
    lgrL = objrLi(i)
    lgTr = objTri(i)
    mbar = 7.d-1 ! average stellar mass in tracer for chapter 2, Jordan, MF,
    ! CMF evaporation prescriptions
    gamma = 2.d-2 ! coefficient in Coulomb logarithm - that of multimass
    ! cluster according to Heggie
    ei = objecci(i)
    Mi = 1.d1**objlgMi(i)
    rhoj0 = objrhoji(i)
    rhoh0 = objrhohi(i)
    W0 = objW0i(i)

    CALL odeint(ystart,nvar,xi,xf,eps,hl,hadv,nok,nbad,orbint,
*      rkqs)
    lgEpt = log10(phiofr(ystart(1))+5.d-1*((1.d1**((2.d0*(
*      ystart(1)-xf)))*(ystart(2)**2.d0)+(1.d1**((2.d0*
*      ystart(3))))))
    logLpt = ystart(1) + ystart(3)

    if (abort.eqv..false.) then
      objrgcf(iok) = ystart(1)

```

```

objvrf(iok) = (1.d1**ystart(1)-xf))*ystart(2)

gamma = atan(objvphii(i)/objvthetai(i))
! due to spherical symmetry, orbital inclination will not
! change even with dynamical friction (i.e. gamma = const)
objvphif(iok) = sign((1.d1**ystart(3))*cos(gamma),
*                   objvphii(i))
objvthetaf(iok) = sign((1.d1**ystart(3))*sin(gamma),
*                   objvthetai(i))

objlgMf(iok) = ystart(4)
objrhohf(iok) = ystart(5)
objEf(iok) = lgEpt
objhf(iok) = (1.d1**logLpt)/LcofE(lgEpt)
objLf(iok) = logLpt
objrpf(iok) = opfn(lgEpt,logLpt,1)
objraf(iok) = opfn(lgEpt,logLpt,4)
objTrf(iok) = opfn(lgEpt,logLpt,5)
objeccf(iok) = (1.d0-1.d1**(objrpf(i)-objraf(i)))/
*             (1.d0+1.d1**(objrpf(i)-objraf(i)))
objrhojf(iok) = log10(rhojev(objrgcf(i),lgEpt,logLpt))

okobjLi(iok) = log10(objLi(i))
okobjEi(iok) = log10(objEi(i))
okobjrgci(iok) = log10(objrgci(i))
okobjlgMi(iok) = objlgMi(i)
okobjrhohi(iok) = objrhohi(i)
okobjrhoji(iok) = objrhoji(i)
okobjhi(iok) = objhi(i)
okobjvri(iok) = objvri(i)
okobjvphii(iok) = objvphii(i)
okobjvthetai(iok) = objvthetai(i)
okobjrpi(iok) = objrpi(i)
okobjrai(iok) = objrai(i)
okobjecci(iok) = objecci(i)
okobjTri(iok) = objTri(i)
iok = iok + 1
end if
end if
end do

OPEN(7,file=evlfilename)
WRITE(7,1000) '#column1:_logarithmic_initial_mass'
WRITE(7,1000) '#column2:_logarithmic_final_mass'
WRITE(7,1000) '#column3:_logarithmic_initial_half-mass_density'
WRITE(7,1000) '#column4:_logarithmic_final_half-mass_density'
WRITE(7,1000) '#column5:_logarithmic_initial_tidal_density'
WRITE(7,1000) '#column6:_logarithmic_final_tidal_density'
WRITE(7,1001) '#column7:_logarithmic_initial_galactocentric_',
*             'radius'
WRITE(7,1000) '#column8:_logarithmic_final_galactocentric_radius'
WRITE(7,1000) '#column9:_logarithmic_initial_orbital_energy'
WRITE(7,1000) '#column10:_logarithmic_final_orbital_energy'
WRITE(7,1000) '#column11:_logarithmic_initial_angular_momentum'
WRITE(7,1000) '#column12:_logarithmic_final_angular_momentum'
WRITE(7,1000) '#column13:_dimensionless_initial_circularity'
WRITE(7,1000) '#column14:_dimensionless_final_circularity'
WRITE(7,1000) '#column15:_dimensionless_initial_radial_velocity'
WRITE(7,1000) '#column16:_dimensionless_final_radial_velocity'
WRITE(7,1001) '#column17:_dimensionless_initial_azimuthal_',
*             'velocity'
WRITE(7,1000) '#column18:_dimensionless_final_azimuthal_velocity'
WRITE(7,1000) '#column19:_dimensionless_initial_polar_velocity'
WRITE(7,1000) '#column20:_dimensionless_final_polar_velocity'
WRITE(7,1000) '#column21:_logarithmic_initial_orbital_pericentre'
WRITE(7,1000) '#column22:_logarithmic_final_orbital_pericentre'
WRITE(7,1000) '#column23:_logarithmic_initial_orbital_pericentre'
WRITE(7,1000) '#column24:_logarithmic_final_orbital_pericentre'
WRITE(7,1000) '#column25:_initial_orbital_ellipticity'
WRITE(7,1000) '#column26:_final_orbital_ellipticity'
WRITE(7,1000) '#column27:_logarithmic_initial_radial_period'
WRITE(7,1000) '#column28:_logarithmic_final_radial_period'

do i = 1,reqnum
  WRITE(7,1009) okobjlgMi(i),objlgMf(i),okobjrhohi(i),objrhohf(i),
*             okobjrhoji(i),objrhojf(i),okobjrgci(i),objrgcf(i),
*             okobjEi(i),objEf(i),okobjLi(i),objLf(i),
*             okobjhi(i),objhf(i),okobjvri(i),objvrf(i),
*             okobjvphii(i),objvphif(i),okobjvthetai(i),
*             objvthetaf(i),okobjrpi(i),objrpf(i),okobjrai(i),
*             objraf(i),okobjecci(i),objeccf(i),okobjTri(i),
*             objTrf(i)
end do
CLOSE(7)

```

```

CALL CPU_TIME(endtime)
print *, 'time_taken_',endtime-starttime
print *, splinterrn, 'interpolation_errors_',check_fort.31'

STOP
301 FORMAT(A,2X,3(E25.14,2X))
302 FORMAT(2(E25.14,2X))
303 FORMAT(2(I4,2X),12(F12.8,2X))
304 FORMAT(A,2X,A,3X,A,6X,A,8X,A,8X,A,8X,A,5X,A,5X,A,3X,A,7X,A,6X,A,
*      2X,A,3X,A,1X,A)
1000 FORMAT(A)
1001 FORMAT(2A)
1002 FORMAT(2(F30.15,2X))
1003 FORMAT(8(E30.10,2X))
6666 FORMAT(12(E30.15,2X))
1004 FORMAT(2A,2(F12.6,A))
1005 FORMAT(3(F30.15,2X))
1006 FORMAT(A,8X,A,27X,A,25X,A,17X,A,15X,A,26X,A,18X,A,27X,A,19X,A,25X,
*      A,25X,A,25X,A)
1007 FORMAT(4A)
1008 FORMAT(8(F30.15,2X))
1009 FORMAT(28(E30.10,2X))
1010 FORMAT(A,12X,A,27X,A,24X,A,25X,A,15X,A,21X,A,24X,A,20X,A)
6004 FORMAT(F30.15)
END

! component function of GHO 1999 point mass correction
SUBROUTINE I0int(x,y,dydx)
IMPLICIT NONE
INTEGER modnum
PARAMETER (modnum=1000)
REAL*8 x,y(*),dydx(*),rppt,Mgaltot,zeta,Mofr
COMMON /pmfixpath/ rppt,Mgaltot
zeta = (x**2.d0+1.d0)**(5.d-1)
dydx(1) = Mofr(rppt*zeta)/(Mgaltot*(zeta**3.d0))
return
END

! component function of GHO 1999 point mass correction
SUBROUTINE I1int(x,y,dydx)
IMPLICIT NONE
INTEGER modnum
PARAMETER (modnum=1000)
REAL*8 x,y(*),dydx(*),rppt,Mgaltot,zeta,dMdnrofr
COMMON /pmfixpath/ rppt,Mgaltot
zeta = (x**2.d0+1.d0)**(5.d-1)
dydx(1) = dMdnrofr(rppt*zeta)/(Mgaltot*(zeta**3.d0))
return
END

! component function of GHO 1999 point mass correction
SUBROUTINE J0int(x,y,dydx)
IMPLICIT NONE
INTEGER modnum
PARAMETER (modnum=1000)
REAL*8 x,y(*),dydx(*),rppt,Mgaltot,zeta,Mofr
COMMON /pmfixpath/ rppt,Mgaltot
zeta = (x**2.d0+1.d0)**(5.d-1)
dydx(1) = Mofr(rppt*zeta)/(Mgaltot*(zeta**5.d0))
return
END

! component function of GHO 1999 point mass correction
SUBROUTINE J1int(x,y,dydx)
IMPLICIT NONE
INTEGER modnum
PARAMETER (modnum=1000)
REAL*8 x,y(*),dydx(*),rppt,Mgaltot,zeta,dMdnrofr
COMMON /pmfixpath/ rppt,Mgaltot
zeta = (x**2.d0+1.d0)**(5.d-1)
dydx(1) = dMdnrofr(rppt*zeta)/(Mgaltot*(zeta**5.d0))
return
END

! mass of host galaxy interior to radius r
FUNCTION Mofr(r)
IMPLICIT NONE
REAL*8 Mofr,r,dphidrofr
Mofr = (r**2.d0)*dphidrofr(log10(r))
return
END

```

```

! logarithmic derivative of host galaxy mass interior to radius r at r
FUNCTION dMdlnofr(r)
IMPLICIT NONE
REAL*8 dMdlnofr,r,Mofr,d2phidr2ofr
dMdlnofr = 2.d0*Mofr(r)+(r**3.d0)*d2phidr2ofr(log10(r))
return
END

! Gnedin, Hernquist & Ostriker 1999 tidal shock point mass correction function
FUNCTION chi(lgrp)
IMPLICIT NONE
INTEGER shokn
PARAMETER (shokn=500)
REAL*8 lgchi(shokn),shokrp(shokn),d2chidrp2(shokn),lgchipt,lgrp
REAL*8 chi
CHARACTER*30 splintcall
COMMON /chipath/ lgchi,shokrp,d2chidrp2
COMMON /splintpath/ splintcall
splintcall = 'chi_shokrp'
CALL splintcheck(shokrp,lgchi,d2chidrp2,shokn,lgrp,lgchipt,.true.)
chi = 1.d1**lgchipt
return
END

! Weinberg 1994 tidal shock adiabatic correction function
FUNCTION adiab(lgrhoh,lgrp,lgvp)
IMPLICIT NONE
REAL*8 adiab,lgrhoh,lgrp,lgvp,P0,G,m0overM0
COMMON /parampath/ P0,G,m0overM0
adiab = (1.d0+(P0/3.d0)*1.d1**((lgrhoh+2.d0*(lgrp-lgvp)))*(-1.5d0))
return
END

! Aguilar, Hutt & Ostriker 1988 tidal field time variance shock correction function
FUNCTION lambda(lgrp,lgra)
IMPLICIT NONE
REAL*8 lambda,rp,ra,Mofr,lgrp,lgra
rp = 1.d1**lgrp
ra = 1.d1**lgra
lambda = (1.d0-(Mofr(ra)/Mofr(rp))*((rp/ra)**3.d0))**2.d0
return
END

! system of ODEs for cumulative King concentration distribution integration
SUBROUTINE cint(x,y,dydx)
IMPLICIT NONE
REAL*8 x,y(*),dydx(*),cdist
dydx(1) = cdist(x)
return
END

! King concentration distribution function
FUNCTION cdist(c)
IMPLICIT NONE
INTEGER conctype
REAL*8 cdist,c,rho0param1,rho0param2,gaussian,schechter
COMMON /rho0path/ rho0param1,rho0param2,conctype
if (conctype.eq.1) cdist = gaussian(c,rho0param1,rho0param2)
if (conctype.eq.2) cdist = schechter(c,rho0param1,rho0param2)
return
END

! Gaussian with mean (mu) and standard deviation (sig)
FUNCTION gaussian(x,mu,sig)
IMPLICIT NONE
REAL*8 gaussian,x,mu,sig
gaussian = exp(-((x-mu)**2.d0)/(2.d0*(sig**2.d0)))
return
END

! Schechter function with power-law slope (-beta) and exponential cut-off (xc)
FUNCTION schechter(x,beta,xc)
IMPLICIT NONE
REAL*8 schechter,x,beta,xc
schechter = (x**(-beta))*exp(-x/xc)
return
END

! system of ODEs for radius as a function of azimuth
SUBROUTINE rofpsiint(x,y,dydx)
IMPLICIT NONE
REAL*8 logE,logL,x,y(*),dydx(*),r,psi,dphidrofr,hpt,LcofE
COMMON /ELpath/ logE,hpt

```

```

r = 1.d1**y(1)
psi = 1.d1**x
logL = log10(LcofE(logE)*hpt)
dydx(1) = y(2)
dydx(2) = log(1.d1)*(1.d1**(2.d0*x))*(1.d0-dphidrofr(y(1))*
* (1.d1**(3.d0*y(1)-2.d0*logL)))+y(2)*(1.d0+y(2))
return
END

! system of ODEs for radial/azimuthal period
SUBROUTINE azint(x,y,dydx)
IMPLICIT NONE
INTEGER rofpsin
PARAMETER (rofpsin=200)
REAL*8 x,y(*),dydx(*),rofpsi(rofpsin),psi(rofpsin)
REAL*8 spd2rdpsi2(rofpsin),rpt
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /azpath/ rofpsi,psi,spd2rdpsi2
splintcall = 'azint_psi-rofpsi'
CALL splintcheck(psi,rofpsi,spd2rdpsi2,rofpsin,x,rpt,.true.)
dydx(1) = 1.d1**(2.d0*rpt+x)
return
END

! system of ODEs for time averaged orbital radius
SUBROUTINE rbarint(x,y,dydx)
IMPLICIT NONE
INTEGER rofpsin
PARAMETER (rofpsin=200)
REAL*8 x,y(*),dydx(*),rofpsi(rofpsin),psi(rofpsin)
REAL*8 spd2rdpsi2(rofpsin),rpt
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /azpath/ rofpsi,psi,spd2rdpsi2
splintcall = 'rbarint_psi-rofpsi'
CALL splintcheck(psi,rofpsi,spd2rdpsi2,rofpsin,x,rpt,.true.)
dydx(1) = 1.d1**(3.d0*rpt+x)
return
END

! system of ODEs for time averaged tidal density
SUBROUTINE jbarint(x,y,dydx)
IMPLICIT NONE
INTEGER rofpsin
PARAMETER (rofpsin=200)
REAL*8 x,y(*),dydx(*),rofpsi(rofpsin),psi(rofpsin),logE,logL
REAL*8 spd2rdpsi2(rofpsin),rpt,rhotid,hpt,LcofE,rhojpower
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /azpath/ rofpsi,psi,spd2rdpsi2
COMMON /rhojint/ rhojpower
COMMON /ELpath/ logE,hpt
logL = log10(hpt*LcofE(logE))
splintcall = 'jbarint_psi-rofpsi'
CALL splintcheck(psi,rofpsi,spd2rdpsi2,rofpsin,x,rpt,.true.)
dydx(1) = (rhotid(rpt,logL)**rhojpower)*1.d1**(2.d0*rpt+x)
return
END

! system of 5 ODEs giving radius, radial and tangential velocities,
! mass, and density
SUBROUTINE orbint(x,y,dydx)
IMPLICIT NONE
REAL*8 x,y(*),dydx(*),dynfric,p0,dphidrofr,df
REAL*8 dlzMevapdlgt,dlzMshokdlgt,dlzMstevdlgt,phiofr,v,lgrp,lgra
REAL*8 v2,rmin,t,lgTr,rhojev,dlgrhohdlgt,Nstar,NofMt
REAL*8 lgEpt,logLpt,opfn,lgrhoj,G,m0overM0,Esup,Lmin,Lmax,rmax
REAL*8 mbar,Mi,rhoj0,rhoh0,W0,gamma,ei,Einf,tf
LOGICAL dead,abort,Lbad,rbad,Ebad,Mbad,rhobad
COMMON /abortpath/ tf,abort
COMMON /parampath/ p0,G,m0overM0
COMMON /deadpath/ rmin,rmax,Lmin,Lmax,Esup,Einf
COMMON /clusterparams/ mbar,Mi,rhoj0,rhoh0,W0,gamma,ei
! y(1) is log rgc
! y(2) is d log rgc/d log t
! y(3) is log vt
! y(4) is log Mc
! y(5) is log rho_h
dead = .false.
rbad = .false.
Lbad = .false.
Ebad = .false.

```

```

Mbad = .false.
rhobad = .false.
v2=(1.d1**(2.d0*(y(1)-x)))*(y(2)**2.d0)+(1.d1**(2.d0*y(3))) ! cluster speed squared
lgEpt = log10(phiofr(y(1))+5.d-1*v2) ! orbital energy
v = v2**5.d-1
logLpt = y(1) + y(3) ! orbital angular momentum
t = 1.d1**x
Nstar = NofMt(y(4),x)
! if the cluster is dead, prevent any further calculations
if (Nstar*gamma.lt.1.d0.or.(y(1).lt.rmin.and.logLpt.lt.Lmin))then
  dead = .true.
  y(4) = -4.d1
else ! if the cluster is not dead, is an abort due to unphysical parameters required?
  if (y(1).ne.y(1).or.y(2).ne.y(2).or.y(3).ne.y(3).or.
*      y(4).ne.y(4).or.y(5).ne.y(5)) then
*      print *, 'y1=',y(1), 'y2=',y(2), 'y3=',y(3), 'y4=',y(4),
*      'y5=',y(5)
*      pause 'NAN_in_orbint'
  end if
  if (logLpt.gt.Lmax.or.logLpt.lt.Lmin) Lbad = .true.
  if (y(1).lt.rmin.or.y(1).gt.rmax) rbad = .true.
  if (lgEpt.gt.Esup.or.lgEpt.lt.Einf.or.lgEpt.ne.lgEpt)Ebad=.true.
  if (1.d1*(-y(4)).eq.0.d0) Mbad = .true.
  if (1.d1*(-y(5)).eq.0.d0) rhobad = .true.
  if (Lbad.or.rbad.or.Ebad.or.Mbad.or.rhobad) then
    abort = .true. ! if trial steps have resulted in unphysical
                    ! parameters, send signal to abort
  return
end if
end if

if (dead) then ! do not proceed with orbit calculations
  dydx(1) = 0.d0 ! d log rgc/d log t
  dydx(2) = 0.d0 ! d2 log rgc/d log t2
  dydx(3) = 0.d0 ! d log vt/d log t
  dydx(4) = 0.d0 ! d log M/d log t
  dydx(5) = 0.d0 ! d log rho-h/d log t
else
  df = dynfric(y(1),y(4),y(5),v)
  lgrp = opfn(lgEpt,logLpt,1)
  lgra = opfn(lgEpt,logLpt,4)
  lgTr = opfn(lgEpt,logLpt,5)
  lgrhoj = log10(rhojev(y(1),lgEpt,logLpt))
  dydx(1) = y(2)
  dydx(2) = (1.d1**(2.d0*x-y(1)))*log(1.d1)*((1.d1**(2.d0*y(3))-
*      y(1))-dphidrof(y(1)))-log(1.d1)*y(2)*(y(2)-1.d0+df*t/v)
  dydx(3) = (df*t/v)-y(2)
  dydx(4) = dlzMevapdlgt(y(1),y(4),y(5),lgrhoj,x)
*      +dlzMshokdlgt(lgrp,lgra,logLpt-lgrp,lgTr,y(5),x)
*      +dlzMstevdlgt(x,y(4),Mi)
  dydx(5) = dlgrhohdlgt(x,dydx(4),Nstar,lgrhoj-y(5))
return
END

! rate of mass-loss due to evaporation
FUNCTION dlzMevapdlgt(lgr,logM,lgrhoh,lgrhoj,logt)
IMPLICIT NONE
REAL PI
PARAMETER (PI=3.141592654d0)
INTEGER evaptype
REAL*8 dlzMevapdlgt,lgr,M,lgrhoj,p0,G,m0overM0,a,b,c,coef,Mi,rhoh
REAL*8 rhoh0,rhoj0,gamma,rtrh,x,t1000,f,omegac2ofr,rhoofr,eta,mu
REAL*8 tref,trh,tcc,t,q,F5,F7,KingF,W0,mbar,ei,lgrhoh
REAL*8 rhotofW0,beta,NofMt,logM,logt,Coullog,Nstar,rhoj
LOGICAL Rfil
COMMON /evappath/ evaptype
COMMON /parampath/ p0,G,m0overM0
COMMON /clusterparams/ mbar,Mi,rhoj0,rhoh0,W0,gamma,ei
Nstar = NofMt(logM,logt)
Coullog = log(Nstar*gamma)
t = 1.d1**logt
M = 1.d1**logM
rhoj = 1.d1**lgrhoj
rhoh = 1.d1**lgrhoh
if (evaptype.eq.0) then
  dlzMevapdlgt = 0.d0
else if (evaptype.eq.1) then ! King 1966
  dlzMevapdlgt = -(27.d0*t/(8.d0*M))*((P0*rhoj/(6.d0*PI))**
*      (5.d-1))*log(5.d-1*Nstar)*KingF(W0)
else if (evaptype.eq.2) then ! Vesperini & Heggie 1997
  dlzMevapdlgt = -(92.d0*t/(75.d0*M))*log(Nstar)*((P0*rhoj)
*      *(5.d-1))
else if (evaptype.eq.3) then ! Fall & Zhang 2001

```

```

    dlzMevapdlgt = -12.105*log(1.2d1)*(t/M)*((P0*rhoj/(4.d0*PI))
*      *(5.d-1))
*   else if (evaptype.eq.4) then ! Jordan et al 2007
    dlzMevapdlgt = -(8.4d2*t/(M*mbar))*((P0*rhoj/(8.d0*PI*G))
*      *(5.d-1))
*   else if (evaptype.eq.5) then ! Chapter 2
    dlzMevapdlgt = -(8.1d2*t/(M*mbar))*((P0*rhoj/(8.d0*PI*G))
*      *(5.d-1))
*   else if (evaptype.eq.6) then ! McLaughlin & Fall 2008
    dlzMevapdlgt = -(1.1d3*t/(M*mbar))*((P0*rhoj/(8.d0*PI*G))
*      *(5.d-1))
*   else if (evaptype.eq.7) then ! Chandar et al 2007
    dlzMevapdlgt = -(5.6d2*t/(M*mbar))*((P0*rhoj/(8.d0*PI*G))
*      *(5.d-1))
*   else if (evaptype.eq.8) then ! Baumgardt 1998
    dlzMevapdlgt = -(8.d0*t/(M*69.d0))*(((P0/3.d0)*(2.d0*rhoj+
*      (14.9**3.d0)*rhoj))**((5.d-1))*Coullog
*   else if (evaptype.eq.9) then ! Gieles & Baumgardt 2008
    dlzMevapdlgt = -(3.7d1*t/(M*6.9d2))*exp(1.d1*((rhoj/(2.d0*rhoj)
*      )*(1.d0/3.d0))*Coullog*((2.d0*P0*rhoj/3.d0)**(5.d-1))
*   else if (evaptype.eq.10) then ! Gieles et al 2011
    dlzMevapdlgt = -2.d1*(t/M)*((P0*rhoj/(8.d0*PI*(0.145**3.d0)))
*      *(5.d-1))
*   else if (evaptype.eq.11) then ! Gieles & Baumgardt 2008 non-linear
    dlzMevapdlgt = -7.4d-3*(t/M)*exp(1.d1*((rhoj/(2.d0*rhoj))*((
*      1.d0/3.d0))*((Coullog/1.38d-1)**(7.5d-1))*((M/
*      3.85d0)**(2.5d-1))*((2.d0*P0*rhoj/3.d0)**(5.d-1))
*   else if (evaptype.eq.12) then ! Gieles et al 2011 non-linear
    x = 7.5d-1
    dlzMevapdlgt = -2.d1*(t/M)*((P0*rhoj/(8.d0*PI*(0.145**3.d0)))
*      *(5.d-1))*((M*1.d-5)**(1.d0-x))
*   else if (evaptype.eq.13) then ! Tanikaway & Fukushige 2010
    rtrh = (2.d0*1.d1*(rhoj0-rhoj0))*((1.d0/3.d0)
    t1000 = 2.d4*((rtrh**4.d0)/((rtrh**4.d0)+2.d3))
    x = (rtrh**4.d0)/((rtrh**4.d0)+1.d2)
    if (rtrh.gt.8.d0) then ! function g is built in using if statement
      f = 0.d0
    else
      f = 15.d0/(16.d0-((1.d0+P0*rhoj0)/(omegac2ofr(lgr))**2.d0))
    end if
    dlzMevapdlgt = -Mi*(t/M)*((P0*rhoj0)**(5.d-1))/(2.54d2*t1000*(
*      1.d0+f)*((1.5d-5*Mi)**(x)))
*   else if (evaptype.eq.14) then ! Lamers et al 2010
    ! if current tidal density is greater or equal to initial tidal
    ! density, then the cluster must be filling its Roche lobe
    if (lgrrhoj.ge.rhoj0) Rfil = .true.
    trh = 0.169d0*M/((P0*rhoj0)**(5.d-1))*Coullog
    if (Rfil) then
      tcc = 16.9d0*(trh**(-0.872d0))
      if (t.lt.tcc) then ! pre core-collapse
        if (abs(W0-7.d0).lt.abs(W0-5.d0)) then ! W0 is closer to 7
          eta = 0.8d0
          mu = 0.8d0
          tref = 3.5d0
          a = 1.23d0
          b = -0.094d0
          c = 0.d0
          coef = 1.d0
        else ! W0 is closer to 5
          eta = 0.65d0
          mu = 0.65d0
          tref = 13.3d0
          a = 1.528d0
          b = -0.121d0
          c = 0.d0
          coef = 1.d0
        end if
      else ! post core-collapse
        if (abs(W0-7.d0).lt.abs(W0-5.d0)) then
          mu = 0.8d0
          tref = 6.2d0
          a = 0.893d0
          b = -0.0691d0
          c = 0.d0
        else
          mu = 0.65d0
          tref = 7.2d0
          a = 1.528d0
          b = -0.0984d0
          c = 0.d0
        end if
      if (M.lt.1.d3) then
        ! mass-loss depends on cluster mass in post core-collapse regime

```



```

        eta = 0.4d0
        coef = ((1.d1)**(0.9d0))
    else
        eta = 0.7d0
        coef = 1.d0
    end if
end if
else
    F7 = (rhoj/rhotofW0(7.d0))**(1.d0/3.d0)
    F5 = (rhoj/rhotofW0(5.d0))**(1.d0/3.d0)
    tcc = 32.d0*(trh**(-0.872d0))*(F5**(-0.513d0))
    if (t.lt.tcc) then ! pre core-collapse
        eta = 0.8d0
        mu = 0.8d0
        a = 1.38
        b = -0.0984d0
        c = 0.101d0
        coef = 1.d0
        if (F7.gt.-0.5) then
            tref = 3.5d0
        else
            tref = 7.395d0*(F7**(0.65d0))
        end if
    else ! post core-collapse
        a = 1.38d0
        b = -0.0984d0
        c = 0.101d0
        if (F7.gt.-0.5) then
            tref = 3.5d0
        else
            tref = 7.395d0*(F7**(0.65d0))
        end if
        if (M.lt.1.d3) then
            eta = 0.4d0
            coef = ((1.d1)**(0.9d0))
        else
            eta = 0.7d0
            coef = 1.d0
        end if
    end if
end if
if (t.gt.tcc.and.Rfil.eqv..false.) then
    q = (1.d0+b*mu)**(-1.d0)
    dlzMevapdlgt = -((1.507d0**eta)*(a**(q*mu))*(Mi**(-0.0984*eta
*      *mu+b*q*(mu**2.d0)))/(7.5d0*coef*(F5**0.127d0)*(F7
*      *(0.207d0+q*c*mu))*(9.d0*tref*(1.d0-ei)/3.4d2)**
*      q))*(t/M)*(M**(1.d0-eta))*((P0*rhoj/6.d0)**
*      (5.d-1*q))
    else
        q = (1.d0+b*eta)**(-1.d0)
        dlzMevapdlgt = -((a**(q*eta))*(Mi**(b*q*eta*mu))/((9.d0*tref*
*      (1.d0-ei)/3.4d2**q))*(t/M)*(M**(1.d0-eta))*((P0*
*      rhoj/6.d0)**(5.d-1*q))
    end if
else if (evaptype.eq.15) then ! Takahashi & Baumgardt 2012
    mu = 40.d0
    beta = 2.d0
    dlzMevapdlgt = -(mu*(M**(1.d0/(beta+2.d0)))/(2.d0*PI))*(t/M)*
*      ((P0*rhoj/3.d0)**(5.d-1))*(((2.d0*rhoj/rhoj)
*      *(5.d-1))*(2.d0*PI*Coulllog/(0.138d0*mu)))
*      *((beta+1.d0)/(beta+2.d0)))
end if
return
END

! King's evaporation coefficient as a function of W0
FUNCTION KingF(W0)
IMPLICIT NONE
REAL*8 KingF,W0,Ftab(9),W0tab(9),spd2FdW02(9)
SAVE Ftab,W0tab,spd2FdW02
DATA Ftab(1)/7.3d0/,Ftab(2)/6.25d0/,Ftab(3)/4.71d0/,
* Ftab(4)/3.83d0/,Ftab(5)/3.59d0/,Ftab(6)/4.06d0/,
* Ftab(7)/5.41d0/,Ftab(8)/7.04d0/,Ftab(9)/7.68d0/
DATA W0tab(1)/2.5d0/,W0tab(2)/3.d0/,W0tab(3)/4.d0/,
* W0tab(4)/5.d0/,W0tab(5)/6.d0/,W0tab(6)/7.d0/,W0tab(7)/8.d0/,
* W0tab(8)/9.d0/,W0tab(9)/1.d1/
DATA spd2FdW02 /9 * 0.d0/
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
splintcall = 'KingF_W0'
if (W0.lt.W0tab(1)) then
    KingF = Ftab(1)
else if (W0.gt.W0tab(9)) then

```

```

      KingF = Ftab(9)
    else
      CALL splintcheck(W0tab,Ftab,spd2FdW02,9,W0,KingF,..true.)
    end if
    return
  END

  ! density inside of King zero density surface as a function of W0
  FUNCTION rhotofW0(W0pt)
  IMPLICIT NONE
  INTEGER kingdatn
  PARAMETER (kingdatn=193)
  REAL*8 kingW0(kingdatn),spd2MdW02(kingdatn)
  REAL*8 kingM(kingdatn),lgrhot,lgW0,rhotofW0
  REAL*8 kingrhot(kingdatn),spd2rhotdW02(kingdatn),W0pt
  CHARACTER*30 splintcall
  COMMON /kingpath/ kingW0,kingM,kingrhot,spd2MdW02,spd2rhotdW02
  COMMON /splintpath/ splintcall
  splintcall = 'kingW0_rhot'
  lgW0 = log10(W0pt)
  CALL splintcheck(kingW0,kingrhot,spd2rhotdW02,kingdatn,lgW0,
*                                     lgrhot,..false.)
  rhotofW0 = 1.d1**lgrhot
  return
  END

  ! mass-loss rate due to tidal shocking
  FUNCTION dlzMshokdlgt(lgrp,lgra,lgvp,lgTr,lgrhoh,logt)
  IMPLICIT NONE
  INTEGER shoktype
  REAL*8 dlzMshokdlgt,lgrp,lgra,lgvp,lgTr,lgrhoh,Mofr
  REAL*8 P0,G,m0overM0,adiab,chi,lambdalogt
  COMMON /shokpath/ shoktype
  COMMON /parampath/ p0,G,m0overM0
  if (shoktype.eq.0) then
    dlzMshokdlgt = 0.d0
  else if (shoktype.eq.1) then
    dlzMshokdlgt = -(8.d1/(3.d0*P0))*1.d1**((logt-lgTr-lgrhoh-2.d0*
* (lgvp+2.d0*lgrp))*(Mofr(1.d1**lgrp)**2.d0)*
* adiab(lgrhoh,lgrp,lgvp)*chi(lgrp)*lambda(lgrp,lgra))
  end if
  return
  END

  ! mass-loss rate due to stellar evolution
  FUNCTION dlzMstevdlgt(logt,logM,Minit)
  IMPLICIT NONE
  INTEGER stevtype
  REAL*8 dlzMstevdlgt,logt,lgmto,mtooft,ChabIMF,dlgmtodlgttoft,Pej
  REAL*8 lgml,lgmu,Pwd,Pns,Pbh,lgmwd,lgmns,mto,Minit,Mnorm,a,b,c,d
  REAL*8 logM,mrm
  COMMON /stevparam/ lgml,lgmu,Pwd,Pns,Pbh,lgmwd,lgmns,Mnorm
  COMMON /stevpath/ stevtype
  if (stevtype.eq.0) then
    dlzMstevdlgt = 0.d0
  else
    mto = mtooft(logt)
    lgmto = log10(mto)
    if (lgmto.gt.lgmu.or.lgmto.lt.lgml) then
      ! not possible to be losing mass due to stellar
      ! evolution when current mass expiring is greater
      ! than biggest mass in IMF or smaller than
      ! smallest mass in IMF
      dlzMstevdlgt = 0.d0
    else
      dlzMstevdlgt = ChabIMF(lgmto)*dlgmtodlgttoft(logt)*(mto-(1.d0
* -Pej(lgmto))*mrm(lgmto))/((1.d1**logM)*log(1.d1))*
* Minit/Mnorm
      a = ChabIMF(lgmto)
      b = dlgmtodlgttoft(logt)
      c = Pej(lgmto)
      d = mrm(lgmto)
    end if
  end if
  return
  END

  ! magnitude of dynamical friction according to Chandrasekhar
  FUNCTION dynfric(lgr,lgM,lgrhoh,v)
  IMPLICIT NONE
  REAL*8 PI
  PARAMETER (PI=3.141592654d0)
  INTEGER frictype

```

```

REAL*8 dynfric,lgr,lgM,lgrrhoh,v,u,vd2ofr,P0,m0overM0,rhoofr
REAL*8 lncoul,error,drhodr,G,dlgrhodlgr
COMMON /parampath/ P0,G,m0overM0
COMMON /fricpath/ frictype
if (frictype.eq.0) then
  dynfric = 0.d0
else if (frictype.eq.1) then
  dlgrhodlgr = ((1.d1**lgr)/rhoofr(lgr))*drhodr(lgr)
  if (-dlgrhodlgr.le.0.1d0) then
    ! if logarithmic density gradient is more shallow than 0.1,
    ! consider density to be constant
    dynfric = 0.d0
  else
    ! this is the usual  $X = v/\sqrt{2}*\sigma$ 
    u = v/(2.d0*vd2ofr(lgr))*5.d-1
    ! Chandrasekhar's formula from BT and made dimensionless
    dynfric = -(P0*m0overM0*(1.d1**lgM)*rhoofr(lgr)*
    * lncoul(lgr,lgrrhoh,lgM,v)/(v**2.d0))*(error(u)-
    * 2.d0*u*exp(-(u**2.d0))/(PI**5.d-1))
  end if
end if
return
END

! Coulomb logarithm according to Just & Penarrubia 2005
FUNCTION lncoul(lgr,objlgrhoh,objlgM,objv)
IMPLICIT NONE
REAL*8 lncoul,lgr,objlgrhoh,objlgM,objv,bmax,rhoofr,drhodr,bmin
REAL*8 vd2ofr,PI,a90
PARAMETER (PI=3.141592654d0)
bmax = min(1.d1**lgr,-rhoofr(lgr)/drhodr(lgr))
bmin = ((3.d0/8.d0*PI)*1.d1**((objlgM-objlgrhoh)))*(1.d0/3.d0)
a90 = (1.d1**objlgM)/(2.d0*vd2ofr(lgr)+(objv**2.d0))
lncoul = log(bmax)-5.d-1*log((bmin**2.d0)+(a90**2.d0))
return
END

! squared isotropic velocity dispersion of host galaxy
FUNCTION vd2ofr(lgrpt)
IMPLICIT NONE
INTEGER modnum
PARAMETER (modnum=1000)
REAL*8 vd2ofr,lgrpt,lgr(modnum),lgvd2(modnum),spd2vd2dr2(modnum)
CHARACTER*30 splintcall
COMMON /lgrpath/ lgr
COMMON /vd2path/ lgvd2,spd2vd2dr2
COMMON /splintpath/ splintcall
splintcall = 'lgr_vd2ofr'
CALL splintcheck(lgr,lgvd2,spd2vd2dr2,modnum,lgrpt,vd2ofr,
* .false.)
vd2ofr = 1.d1**vd2ofr
return
END

! rate of change of half-mass density assuming maintained virial equilibrium
FUNCTION dlgrhohdlt(logt,dlgMdlgt,N,lgrrhojrroh)
IMPLICIT NONE
REAL*8 gamrat,Cw,Cb,zeta,PI
PARAMETER (PI=3.141592654d0,gamrat=2.9586751191d0,Cw=0.45d0)
PARAMETER (zeta=7.8d-2,Cb=0.38d0)
REAL*8 dlgrhohdlt,logt,dlgMdlgt,N,lgrrhojrroh,Coullog,kappa
REAL*8 Mi,rhoj0,rhoh0,W0,gamma,ei,mbar,dlgmbardlgtoft
INTEGER clevtype
COMMON /clusterparams/ mbar,Mi,rhoj0,rhoh0,W0,gamma,ei
COMMON /clevpath/ clevtype
if (clevtype.eq.0) then
  dlgrhohdlt = 0.d0
else if (clevtype.eq.1) then
  Coullog = log(N*gamma)
  kappa = ((8.1d1/2.d0)**(1.d0/3.d0))*(gamrat/Cw)*((3.d0*Cb*zeta
  * Coullog/(0.138d0*PI*N))**0.25d0)*
  * (1.d1**((5.d0/24.d0)*(lgrrhojrroh)))
  dlgrhohdlt=dlgmbardlgtoft(logt)*(3.d0-kappa)+dlgMdlgt*(kappa-
  * 5.d0)
  if (dlgrhohdlt.ne.dlgrhohdlt.or.1.d0/dlgrhohdlt.eq.0.d0)then
    pause 'dlgrhohdlt_broken'
  end if
end if
return
END

! definition effective tidal density
FUNCTION rhojev(lgr,lgEpt,lgLpt)
IMPLICIT NONE

```

```

REAL*8 a,b
PARAMETER (a=0.17d0,b=4.1d0)
REAL*8 rhojev, lgr, lgEpt, lgLpt, e, F, lgrp, lgra, rhotid, opfn
INTEGER rtidtype
COMMON /rtidpath/ rtidtype
if (rtidtype.eq.4) then
  lgrp = opfn(lgEpt,lgLpt,1)
  lgra = opfn(lgEpt,lgLpt,4)
  e = (1.d0-1.d1**(lgrp-lgra))/(1.d0+1.d1**(lgrp-lgra))
  F = (1.d1**(lgr-lgrp)-1.d0)/(1.d1**(lgra-lgrp)-1.d0)
  rhojev = log10(rhotid(lgrp,lgLpt))
*      -3.d0*log10(1.d0+a*F*exp(b*e))
else
  rhojev = rhotid(opfn(lgEpt,lgLpt,rtidtype),lgLpt)
end if
END

! returns orbital parameter specified by paramtype for any E and L
FUNCTION opfn(lgEpt,logLpt,paramtype)
IMPLICIT NONE
INTEGER E2dtabn,l2dtabn
PARAMETER (E2dtabn=20,l2dtabn=80)
INTEGER paramtype,frictype, ipt1, ipt2, jpt1, jpt2, i
REAL*8 lgrptab(3*E2dtabn,3*l2dtabn), lgTrtab(3*E2dtabn,3*l2dtabn)
REAL*8 lgratab(3*E2dtabn,3*l2dtabn), opfn, lgEpt, logLpt, lpt
REAL*8 lgrbartab(3*E2dtabn,3*l2dtabn), smalllofL, small
REAL*8 zbl, ztl, zbr, ztr, zpt1, zpt2, lgjbartab(3*E2dtabn,3*l2dtabn)
REAL*8 E2dtab(3*E2dtabn), l2dtab(3*l2dtabn), lgrp, lgra, lgrbar, lgTr
REAL*8 lgjbar
COMMON /paraminterp/ lgrptab, lgratab, lgrbartab, lgjbartab, lgTrtab,
*      E2dtab, l2dtab
COMMON /orbparampath/ lgrp, lgra, lgrbar, lgjbar, lgTr
COMMON /fricpath/ frictype
! paramtype = 1: log preicentre
! paramtype = 2: log time averaged radius
! paramtype = 3: log radius where sqrt jacobi density equals
!                  time average of sqrt jacobi density
! paramtype = 4: log apocentre
! paramtype = 5: log radial period
if (frictype.eq.1) then
  lpt = log10(smalllofL(logLpt,lgEpt))
  small = 1.d40
  do i = 1,3*E2dtabn ! find iteration of l2dtab nearest desired point
    if (ABS(lpt-l2dtab(i)).lt.small) then
      small = ABS(lpt-l2dtab(i))
      ipt1 = i
    end if
  end do
  ! find l2dtab iteration on opposite side of desired point
  if (l2dtab(ipt1).gt.lpt) then
    ipt2 = ipt1
    ipt1 = ipt1 - 1
  else
    ipt2 = ipt1 + 1
  end if

  small = 1.d40
  ! find iteration of E2dtab nearest desired point
  do i = 1,3*E2dtabn
    if (ABS(lgEpt-E2dtab(i)).lt.small) then
      small = ABS(lgEpt-E2dtab(i))
      jpt1 = i
    end if
  end do
  ! find E2dtab iteration on opposite side of desired point
  if (E2dtab(jpt1).gt.lgEpt) then
    jpt2 = jpt1
    jpt1 = jpt2 - 1
  else
    jpt2 = jpt1 + 1
  end if
  ! extrapolate linearly from last 2 points when lpt is out of range
  if (ipt2.gt.3*l2dtabn) then
    ipt2 = 3*l2dtabn
    ipt1 = ipt2 - 1
  else if (ipt1.lt.1) then
    ipt1 = 1
    ipt2 = ipt1 + 1
  end if
  ! extrapolate linearly from last 2 points when Ept is out of range
  if (jpt2.gt.3*E2dtabn) then
    jpt2 = 3*E2dtabn
    jpt1 = jpt2 - 1

```

```

else if (jpt1.lt.1) then
  jpt1 = 1
  jpt2 = jpt1 + 1
end if

if (paramtype.eq.1) then
  zbl = lgrptab(jpt1, ipt1)
  ztl = lgrptab(jpt2, ipt1)
  zbr = lgrptab(jpt1, ipt2)
  ztr = lgrptab(jpt2, ipt2)
else if (paramtype.eq.2) then
  zbl = lgrbartab(jpt1, ipt1)
  ztl = lgrbartab(jpt2, ipt1)
  zbr = lgrbartab(jpt1, ipt2)
  ztr = lgrbartab(jpt2, ipt2)
else if (paramtype.eq.3) then
  zbl = lgjbartab(jpt1, ipt1)
  ztl = lgjbartab(jpt2, ipt1)
  zbr = lgjbartab(jpt1, ipt2)
  ztr = lgjbartab(jpt2, ipt2)
else if (paramtype.eq.4) then
  zbl = lgratab(jpt1, ipt1)
  ztl = lgratab(jpt2, ipt1)
  zbr = lgratab(jpt1, ipt2)
  ztr = lgratab(jpt2, ipt2)
else if (paramtype.eq.5) then
  zbl = lgTrtab(jpt1, ipt1)
  ztl = lgTrtab(jpt2, ipt1)
  zbr = lgTrtab(jpt1, ipt2)
  ztr = lgTrtab(jpt2, ipt2)
end if

zpt2 = (ztr*(lpt-l2dtab(ipt1))+ztl*(l2dtab(ipt2)-lpt))/
* (l2dtab(ipt2)-l2dtab(ipt1))
zpt1 = (zbr*(lpt-l2dtab(ipt1))+zbl*(l2dtab(ipt2)-lpt))/
* (l2dtab(ipt2)-l2dtab(ipt1))
opfn = (zpt2*(lgEpt-E2dtab(jpt1))+zpt1*(E2dtab(jpt2)-lgEpt))/
* (E2dtab(jpt2)-E2dtab(jpt1))

else
  if (paramtype.eq.1) then
    opfn = lgrp
  else if (paramtype.eq.2) then
    opfn = lgrbar
  else if (paramtype.eq.3) then
    opfn = lgjbar
  else if (paramtype.eq.4) then
    opfn = lgra
  else if (paramtype.eq.5) then
    opfn = lgTr
  end if
end if
if (opfn.ne.opfn.or.1.d0/opfn.eq.0.d0) then
  print *, 'NAN_or_INF_in_opfn'
  print *, opfn, lpt, ipt1, ipt2, lgEpt, jpt1, jpt2, paramtype,
* zbl, zbr, ztl, ztr
end if
return
END

! convert from parameter l to angular momentum L
FUNCTION BigLofl(l, lgEpt)
IMPLICIT NONE
INTEGER E2dtabn
PARAMETER (E2dtabn=20)
REAL*8 BigLofl, l, lgEpt, Lminvec(3*E2dtabn), ELminvec(3*E2dtabn)
REAL*8 spd2Lminde2(3*E2dtabn), Lmin, LcofE
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /Lminpath/ Lminvec, ELminvec, spd2Lminde2
splintcall = 'BigLofl_ELminvec_Lminvec'
CALL splintcheck(ELminvec, Lminvec, spd2Lminde2, 3*E2dtabn, lgEpt,
* Lmin, .false.)
BigLofl = 1*LcofE(lgEpt)+(1.d0-1)*(1.d1**Lmin)
return
END

! convert from angular momentum L to parameter l
FUNCTION smalllofL(logL, lgEpt)
IMPLICIT NONE
INTEGER E2dtabn
PARAMETER (E2dtabn=20)
REAL*8 smalllofL, logL, lgEpt, Lminvec(3*E2dtabn), ELminvec(3*E2dtabn)

```

```

REAL*8 spd2LmindE2(3*E2dtabn),Lmin,LcofE
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /Lminpath/ Lminvec,ELminvec,spd2LmindE2
splintcall = 'smalllofL_ELminvec_Lminvec'
CALL splintcheck(ELminvec,Lminvec,spd2LmindE2,3*E2dtabn,lgEpt,
*                                     Lmin,.false.)
smalllofL = (1.d1**logL-1.d1**Lmin)/(LcofE(lgEpt)-1.d1**Lmin)
return
END

! tidal density at radius r due to host galaxy potential and orbit in question
FUNCTION rhotid(lgr,logLpt)
IMPLICIT NONE
REAL*8 rhotid,omega,p0,d2phidr2,logLpt,alpha,d2phidr2ofr
REAL*8 lgr,G,m0overM0
COMMON /parampath/ p0,G,m0overM0
COMMON /alphapath/ alpha
!alpha = 1.d0 ! prograde co-planar star orbit
!alpha = 0.d0 ! parallel radial star orbit
!alpha = -1.d0 ! retrograde co-planar star orbit
omega = 1.d1**((logLpt-2.d0*lgr)
d2phidr2 = d2phidr2ofr(lgr)
if (((alpha**2.d0)+1.d0)*(omega**2.d0)-d2phidr2.lt.0.d0) then
    rhotid = 0.d0 ! tidal compression stuff (Dekel et al)
else
    rhotid = (3.d0/p0)*(alpha*omega+(((omega**2.d0)*((alpha**2.d0)
*                                     +1.d0)-d2phidr2)**5.d-1))**2.d0
end if
return
END

! angular momentum of a circular orbit with energy E
FUNCTION LcofE(lgEpt)
IMPLICIT NONE
INTEGER modnum,i
PARAMETER (modnum=1000)
REAL*8 LcofE,lgEc(modnum),spd2LcdEc2(modnum),logLc(modnum)
REAL*8 lgEpt,rEfind
CHARACTER*30 splintcall
COMMON /Lcpath/ spd2LcdEc2,logLc,lgEc
COMMON /splintpath/ splintcall
splintcall = 'lgEc_LcofE'
CALL splintcheck(lgEc,logLc,spd2LcdEc2,modnum,lgEpt,LcofE,
*                                     .true.)
LcofE = 1.d1**LcofE
return
END

! radius of a circular orbit with energy E
FUNCTION rEfind(lgEpt)
IMPLICIT NONE
INTEGER modnum
PARAMETER (modnum=1000)
REAL*8 spd2LcdEc2(modnum),logLc(modnum),lgEc(modnum)
REAL*8 lgr(modnum),spd2rcdEc2(modnum),spd2rcdLc2(modnum)
REAL*8 rEfind,lgEpt
CHARACTER*30 splintcall
COMMON /lgrpath/ lgr
COMMON /Ecpath/ spd2rcdEc2,spd2rcdLc2
COMMON /Lcpath/ spd2LcdEc2,logLc,lgEc
COMMON /splintpath/ splintcall
splintcall = 'lgEc_rEfind'
CALL splintcheck(lgEc,lgr,spd2rcdEc2,modnum,lgEpt,rEfind,.false.)
rEfind = 1.d1**rEfind
return
END

! radius of a circular orbit with angular momentum L
FUNCTION rLfind(logLpt)
IMPLICIT NONE
INTEGER modnum
PARAMETER (modnum=1000)
REAL*8 spd2LcdEc2(modnum),logLc(modnum),lgEc(modnum)
REAL*8 lgr(modnum),spd2rcdEc2(modnum),spd2rcdLc2(modnum)
REAL*8 rLfind,logLpt
CHARACTER*30 splintcall
COMMON /lgrpath/ lgr
COMMON /Ecpath/ spd2rcdEc2,spd2rcdLc2
COMMON /Lcpath/ spd2LcdEc2,logLc,lgEc
COMMON /splintpath/ splintcall
splintcall = 'logLc_lgr_rLfind'
CALL splintcheck(logLc,lgr,spd2rcdLc2,modnum,logLpt,rLfind,

```

```

*                                     . false .)
rLfind = 1.d1**rLfind
return
END

! number of stars remaining in a cluster of mass M at time t
FUNCTION NofMt(logM,logt)
IMPLICIT NONE
INTEGER mbarn
PARAMETER (mbarn=500)
REAL*8 logtev(mbarn),logmbar(mbarn),spd2mbardt2(mbarn),logt,logM
REAL*8 NofMt,dlgmbardlgt(mbarn),spd3mbardt3(mbarn)
CHARACTER*30 splintcall
COMMON /mbarpath/ logtev,logmbar,dlgmbardlgt,spd2mbardt2,
*                 spd3mbardt3
COMMON /splintpath/ splintcall
splintcall = 'NofMt_logtev_logmbar'
CALL splintcheck(logtev,logmbar,spd2mbardt2,mbarn,logt,NofMt,
*                                     . false .)

NofMt = 1.d1**(logM-NofMt)
return
END

! logarithmic derivative of average stellar mass w.r.t. time
FUNCTION dlgbardlgtft(logt)
IMPLICIT NONE
INTEGER mbarn
PARAMETER (mbarn=500)
REAL*8 logtev(mbarn),logmbar(mbarn),spd2mbardt2(mbarn),logt
REAL*8 dlgbardlgtft,dlgbardlgt(mbarn),spd3mbardt3(mbarn)
CHARACTER*30 splintcall
COMMON /mbarpath/ logtev,logmbar,dlgbardlgt,spd2mbardt2,
*                 spd3mbardt3
COMMON /splintpath/ splintcall
splintcall = 'dlgbardlgtft_logtev_dlgmbardlgt'
CALL splintcheck(logtev,dlgbardlgt,spd3mbardt3,mbarn,logt,
*                 dlgbardlgtft,. false .)
dlgbardlgtft = 1.d1**dlgbardlgtft
return
END

! returns stellar IMF, dN/dlogm as given by Chabrier 2003
FUNCTION ChabIMF(lgm)
IMPLICIT NONE
REAL*8 ChabIMF,lgm,m
m = 1.d1**lgm
if (lgm.le.0.d0) then
  ChabIMF = 0.158d0*exp(-((lgm-log10(0.079d0))**2.d0)
*                                     /(2.d0*((0.69d0)**2.d0)))
else
  ChabIMF = 4.43d-2*(1.d1**(-1.3d0*lgm))
end if
return
END

! mass of main-sequence star expiring at time t
FUNCTION mtooft(lgtpt)
IMPLICIT NONE
INTEGER stevtype
REAL*8 mtooft,lgtpt,logt
COMMON /stevpath/ stevtype
if (stevtype.eq.1) then
  logt = lgtpt+6.d0
  mtooft = 8.5d-1*exp(7.d-1*((6.4d0**1.3d0)-(logt**1.3d0)))
* -2.8d-1*(logt-9.93d0)-2.d-2
  mtooft = 1.d1**mtooft
! Lamers et al 2010 interpretation of Hurley et al 2000 (lowest metallicity)
else if (stevtype.eq.2) then
  logt = lgtpt-3.d0
  mtooft = 0.2732d0-0.3864d0*logt+5.628d-2*(logt**2.d0)
* +1.524d-2*(logt**3.d0)-5.902d-3*(logt**5.d0)
  mtooft = 1.d1**mtooft
end if
return
END

! derivative of mass of main-sequence star expiring at time t w.r.t. time
FUNCTION dlgmtodlgtft(lgtpt)
IMPLICIT NONE
INTEGER stevtype
REAL*8 dlgmtodlgtft,lgtpt,logt
COMMON /stevpath/ stevtype

```

```

if (stevtype.eq.1) then
  logt = lgtpt+6.d0
  ! obtained by fitting for function and differentiating
  dlgmtodlgtot=-0.7735d0*(logt**0.3d0)*
*   exp(7.d-1*((6.4**1.3d0)-(logt**1.3d0)))-0.28d0
else if (stevtype.eq.2) then
  logt = lgtpt-3.d0
  ! Lamers et al 2010 interpretation of Hurley et al 2000 (lowest metallicity)
  dlgmtodlgtot=-0.3864d0+2.d0*5.628d-2*logt
*   +3.d0*1.524d-2*(logt**2.d0)-5.d0*5.902d-3*(logt**4.d0)
end if
return
END

! mass of remnant remaining after progenitor of mass m expires
FUNCTION mrm(lgm)
IMPLICIT NONE
REAL*8 mrm,m,lgml,lgmwd,lgmns,lgmu,lgm,Pwd,Pns,Pbh,McBAGB,Mnorm
INTEGER stevtype
COMMON /stevpath/ stevtype
COMMON /stevparam/ lgml,lgmu,Pwd,Pns,Pbh,lgmwd,lgmns,Mnorm
m = 1.d1*lgm
McBAGB = 0.d0
! Iben and Renzini 1983 as used by Chernoff & Weinberg 1990
if (stevtype.eq.1) then
  if (lgm.lt.lgmwd) then
    mrm = 0.58d0+0.22d0*(m-1.d0)
  else if (lgm.ge.lgmwd.and.lgm.le.lgmns) then
    mrm = 0.d0
  else if (lgm.gt.lgmns) then
    mrm = 1.4d0
  end if
! Hurley et al. 2000 as used by Lamers et al. 2010
else if (stevtype.eq.2) then
  McBAGB = (4.36d-4*(1.d1**((5.22d0*lgm)))+6.84d-2)**2.5d-1
  if (McBAGB.le.0.8d0) then
    ! this corresponds to stars that have a second
    ! dredge up before becoming white dwarfs
    mrm = McBAGB
  else if (0.8d0.lt.McBAGB.and.McBAGB.le.2.25d0) then
    ! this corresponds to stars that do not have
    ! second dredge up before becoming white dwarfs
    mrm = 0.44d0*McBAGB+0.448d0
  else if (McBAGB.gt.2.25d0) then
    ! this corresponds to stars that are massive enough to
    ! go supernova, and result in either a neutron star or
    ! black hole
    mrm = 1.17d0+9.d-2*max(1.44d0,0.773d0*McBAGB-0.35d0)
  end if
end if
return
END

! probability of remnant being ejected upon expiry of progenitor
FUNCTION Pej(lgm)
IMPLICIT NONE
REAL*8 Pej,lgml,lgmwd,lgmns,lgmu,lgm,Pwd,Pns,Pbh,Mnorm
INTEGER stevtype
COMMON /stevpath/ stevtype
COMMON /stevparam/ lgml,lgmu,Pwd,Pns,Pbh,lgmwd,lgmns,Mnorm
if (lgm.le.lgmwd) then
  Pej = Pwd
else if (lgm.gt.lgmwd.and.lgm.le.lgmns) then
  Pej = Pns
else if (lgm.gt.lgmns) then
  Pej = Pbh
else
  PAUSE 'Pej out of bounds'
end if
return
END

! system of ODEs to give number of stars and remnants remaining at time t
SUBROUTINE SNint(x,y,dydx)
IMPLICIT NONE
REAL*8 x,y(*),dydx(*),ChabIMF,Pej,lgml,lgmu,Pwd,Pns,Pbh
REAL*8 lgmwd,lgmns,Mnorm
LOGICAL initint
COMMON /SNSMpath/ initint
COMMON /stevparam/ lgml,lgmu,Pwd,Pns,Pbh,lgmwd,lgmns,Mnorm
if (initint) then
  dydx(1) = ChabIMF(x)
else

```



```

dydx(1) = ChabIMF(x)*Pej(x)
end if
return
END

! system of ODEs to give mass of stars and remnants remaining at time t
SUBROUTINE SMint(x,y,dydx)
IMPLICIT NONE
REAL*8 x,y(*),dydx(*),ChabIMF,Pej,mm,m,lgml,lgmu,Pwd,Pns,Pbh
REAL*8 lgmwd,lgmns,Mnorm
LOGICAL initint
COMMON /SNSMpath/ initint
COMMON /stevparam/ lgml,lgmu,Pwd,Pns,Pbh,lgmwd,lgmns,Mnorm
m = 1.d1**x
if (initint) then
dydx(1) = ChabIMF(x)*m
else
dydx(1) = ChabIMF(x)*(m-(1.d0-Pej(x))*mm(x))
end if
return
END

! function to find radius at which tidal density equals
! a specific value through bisection
FUNCTION rjbarfunc(lgr)
IMPLICIT NONE
REAL*8 rjbarfunc,lgr,rhojbar,hpt,logE,rhotid,rhojpower,logL,LcofE
COMMON /rjbarpath/ rhojbar
COMMON /rhojint/ rhojpower
COMMON /ELpath/ logE,hpt
logL = log10(LcofE(logE)*hpt)
rjbarfunc = rhojpower*log10(rhotid(lgr,logL))-rhojbar
return
END

! function to find pericentric and apocentric radii through bisection
FUNCTION hfunc(rturn)
IMPLICIT NONE
REAL*8 hfunc,rturn,phiofr,logE,hpt,lgrEpt,rmaxpt,LcofE
COMMON /hfuncpath/ lgrEpt,rmaxpt
COMMON /ELpath/ logE,hpt
if (rturn.eq.lgrEpt) then ! by definition, (E-phi(rE))rE2 = vc2(rE)rE2 = Lc(E)2
hfunc = (LcofE(logE)**2.d0)*(1.d0-hpt**2.d0)
else if (rturn.eq.rmaxpt) then ! by definition, (E-phi(rmaxpt)) = 0
hfunc = -(hpt*LcofE(logE))**2.d0
else
hfunc = 2.d0*((1.d1**logE)-phiofr(rturn))*(1.d1**(2.d0*rturn))-
(hpt*LcofE(logE))**2.d0
*
end if
return
END

! orbital radius as a function of azimuth
FUNCTION psifunc(x)
IMPLICIT NONE
INTEGER rofpsin
PARAMETER (rofpsin=200)
REAL*8 psifunc,x,rofpsi(rofpsin),psi(rofpsin),rpt
REAL*8 spd2rdpsi2(rofpsin)
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /azpath/ rofpsi,psi,spd2rdpsi2
splintcall = 'psifunc_psi_rofpsi'
CALL splintcheck(psi,rofpsi,spd2rdpsi2,rofpsin,x,rpt, .false.)
psifunc = rpt
return
END

! distribution function as a function of energy
FUNCTION isodf(lgEpt)
IMPLICIT NONE
INTEGER mdfnum
PARAMETER (mdfnum=2000)
REAL*8 isodf,lgEpt,mdflgE(mdfnum),mdflgisodf(mdfnum)
REAL*8 spd2fdE2(mdfnum),lgisodfpt
CHARACTER*30 splintcall
COMMON /isodfpath/ mdflgE,mdflgisodf,spd2fdE2
COMMON /splintpath/ splintcall
splintcall = 'isodf_mdflgE_mdflgisodf'
CALL splintcheck(mdflgE,mdflgisodf,spd2fdE2,mdfnum,lgEpt,
*
lgisodfpt, .false.)
isodf = 1.d1**lgisodfpt
return

```

END

```

! radius as a function of host galaxy potential
FUNCTION rofphi(lgphipt)
IMPLICIT NONE
INTEGER modnum
PARAMETER (modnum=1000)
REAL*8 rofphi, lgr(modnum), lgphi(modnum), dlghidlgr(modnum)
REAL*8 d2lgphidlgr2(modnum), spd2phidr2(modnum), spd3phidr3(modnum)
REAL*8 spd4phidr4(modnum), lgphipt, lgrpt, spd2rdphi2(modnum)
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /lgrpath/ lgr
COMMON /phiandr/ lgphi, dlghidlgr, d2lgphidlgr2, spd2phidr2,
*          spd3phidr3, spd4phidr4, spd2rdphi2
* splintcall = 'rofphi_lgphi_lgr'
CALL splintcheck(lgphi, lgr, spd2rdphi2, modnum, lgphipt,
*          lgrpt, .false.)
rofphi = 1.d1**lgrpt
return
END

! host galaxy potential as a function of radius
FUNCTION phiofr(lgrpt)
IMPLICIT NONE
INTEGER modnum, modeltype
REAL*8 PI
PARAMETER (modnum=1000, PI=3.141592654d0)
REAL*8 phiofr, lgrpt, lgr(modnum), lgphi(modnum), dlghidlgr(modnum)
REAL*8 d2lgphidlgr2(modnum), spd2phidr2(modnum), spd3phidr3(modnum)
REAL*8 spd4phidr4(modnum), lgphipt, spd2rdphi2(modnum), k, p0, r
REAL*8 rref, error, G, m0overM0
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /modelpath/ k, rref, modeltype
COMMON /parampath/ p0, G, m0overM0
COMMON /lgrpath/ lgr
COMMON /phiandr/ lgphi, dlghidlgr, d2lgphidlgr2, spd2phidr2,
*          spd3phidr3, spd4phidr4, spd2rdphi2
* r = 1.d1**lgrpt
if (modeltype.eq.1) then ! polytropes
  if (lgrpt.lt.lgr(1)) then ! small r asymptotes
    phiofr = p0*(r**2.d0)/6.d0 -
    * k*(p0**2.d0)*(r**4.d0)/(2.4d2*(k-1.d0)) +
    * k*(3.d0*k+1.d1)*(p0**3.d0)*(r**6.d0)/((6048.d1)*
    * ((k-1.d0)**2.d0)) -
    * k*(9.d0*(k**2.d0)+8.6d1*k+2.8d2)*(p0**4.d0)*(r**8.d0
    * )/(2612736.d1*((k-1.d0)**3.d0))
  else if (lgrpt.gt.lgr(modnum)) then ! large r asymptotes
    if (k.lt.2.d0) then
      phiofr = ((P0/(3.d0-k))*(5.d-1*(2.d0-k)))*((2.d0*(k-1.d0))
    * ** (5.d-1*k))*(1.d1**((2.d0-k)*lgrpt))/(2.d0-k)
    else if (k.gt.2.d0) then
      phiofr = (2.d0*(k-1.d0)/(k-2.d0))*(1.d0-((P0/(2.d0*(k-1.d0)
    * *(3.d0-k)))*(5.d-1*(2.d0-k)))*(1.d1**((2.d0-k)*lgrpt)))
    else
      phiofr = log(5.d-1*P0*(1.d1**((2.d0*lgrpt)))
    end if
  else ! numerical results for inbetween bits
    splintcall = 'phiofr_lgr_lgphi'
    CALL splintcheck(lgr, lgphi, spd2phidr2, modnum, lgrpt,
  *          lgphipt, .false.)
  phiofr = 1.d1**lgphipt
end if
else if (modeltype.eq.2) then ! king models
  if (lgrpt.lt.lgr(1)) then ! small r asymptotes
    phiofr = p0*(r**2.d0)/6.d0 + (r**4.d0)*
  * (p0**2.d0)*(2.d0*((k/PI)**5.d-1)-exp(k)*error(k**5.d-1))/1.2d2
  else ! numerical results for everything between small r and r_t
    splintcall = 'phiofr_lgr_lgphi'
    CALL splintcheck(lgr, lgphi, spd2phidr2, modnum, lgrpt,
  *          lgphipt, .false.)
  phiofr = 1.d1**lgphipt
end if
else if (modeltype.eq.3) then ! dehn models
  if (k.eq.2.d0) then
    phiofr = 4.d0*p0*log(r*(rref+1.d0)/((r+1.d0)*rref))
  else
    phiofr = ((2.d0*(4.d0-k))*p0/(3.d0-k))*(((r/(r+1.d0))
  * ** (2.d0-k))-((rref/(rref+1.d0))** (2.d0-k)))
  end if
else if (modeltype.eq.4) then ! powerlaw models
  if (k.eq.2.d0) then

```

```

        phiofr = p0*log(r/rref)
    else
        phiofr = (p0/((3.d0-k)*(2.d0-k)))*
*         (r**((2.d0-k)-rref**((2.d0-k)))
    end if
end if
return
END

! first derivative of host galaxy potential w.r.t. radius
FUNCTION dphidrof(lgrpt)
IMPLICIT NONE
INTEGER modnum,modeltype
REAL*8 PI
PARAMETER (modnum=1000,PI=3.141592654d0)
REAL*8 dphidrof,lgr(modnum),lgphi(modnum),dlgphidlgr(modnum)
REAL*8 d2lgphidlgr2(modnum),spd2phidr2(modnum),spd3phidr3(modnum)
REAL*8 spd4phidr4(modnum),dlgphidlgrpt,spd2rdphi2(modnum),k,r,p0
REAL*8 rref,error,G,m0overM0,lgrpt,phiofr
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /modelpath/ k,rref,modeltype
COMMON /parampath/ p0,G,m0overM0
COMMON /lgrpath/ lgr
COMMON /phiandr/ lgphi,dlgphidlgr,d2lgphidlgr2,spd2phidr2,
*         spd3phidr3,spd4phidr4,spd2rdphi2
r = 1.dl**lgrpt
if (modeltype.eq.1) then ! polytropes
    if (lgrpt.lt.lgr(1)) then ! small r asymptotes
        dphidrof = p0*r/3.d0 -
*         k*(p0**2.d0)*(r**3.d0)/(6.d1*(k-1.d0)) +
*         k*(3.d0*k+1.d1)*(p0**3.d0)*(r**5.d0)/((1008.d1)
*         *((k-1.d0)**2.d0)) -
*         k*(9.d0*(k**2.d0)+8.6d1*k+2.8d2)*(p0**4.d0)*
*         (r**7.d0)/(326592.d1*((k-1.d0)**3.d0))
    else if (lgrpt.gt.lgr(modnum)) then ! large r asymptotes
        if (k.lt.2.d0) then
            dphidrof = ((P0/(3.d0-k))*(5.d-1*(2.d0-k)))*((2.d0*
*             (k-1.d0))*(5.d-1*k))*(1.d1**((1.d0-k)*lgrpt))
        else if (k.gt.2.d0) then
            dphidrof = 2.d0*(k-1.d0)*((P0/(2.d0*(k-1.d0)*(3.d0-k)))
*             *(5.d-1*(2.d0-k)))*(1.d1**((1.d0-k)*lgrpt))
        else
            dphidrof = 2.d0*(1.d1*(-lgrpt))
        end if
    else ! numerical results for inbetween bits
        splintcall = 'dphidrof_lgr_dlgphidlgr'
        CALL splintcheck(lgr,dlgphidlgr,spd3phidr3,modnum,lgrpt,
*         dlgphidlgrpt,.false.)
        dphidrof = (1.dl**((log10(phiofr(lgrpt))-lgrpt))*dlgphidlgrpt)
    end if
else if (modeltype.eq.2) then ! king models
    if (lgrpt.lt.lgr(1)) then ! small r expansions
        dphidrof = p0*r/3.d0 + (r**3.d0)*(p0**2.d0)
*         *(2.d0*((k/PI)**5.d-1)-exp(k)*error(k**5.d-1))/3.d1
    else ! numerical results for everything between small r and r_t
        splintcall = 'dphidrof_lgr_dlgphidlgr'
        CALL splintcheck(lgr,dlgphidlgr,spd3phidr3,modnum,lgrpt,
*         dlgphidlgrpt,.false.)
        dphidrof = (1.dl**((log10(phiofr(lgrpt))-lgrpt))*dlgphidlgrpt)
    end if
else if (modeltype.eq.3) then ! dehn models
    dphidrof = (2.d0**((4.d0-k))*p0*(r**((1.d0-k)))*
*         ((r+1.d0)**(k-3.d0))/(3.d0-k)
else if (modeltype.eq.4) then ! powerlaw models
    dphidrof = p0*(r**((1.d0-k)))/(3.d0-k)
end if
return
END

! second derivative of host galaxy potential w.r.t. radius
FUNCTION d2phidr2of(lgrpt)
IMPLICIT NONE
INTEGER modnum,modeltype
REAL*8 PI
PARAMETER (modnum=1000,PI=3.141592654d0)
REAL*8 d2phidr2of,lgr(modnum),lgphi(modnum),dlgphidlgr(modnum)
REAL*8 d2lgphidlgr2(modnum),spd2phidr2(modnum),spd3phidr3(modnum)
REAL*8 spd4phidr4(modnum),d2lgphidlgr2pt,spd2rdphi2(modnum),k,p0
REAL*8 rref,error,r,G,m0overM0,lgrpt,dlgphidlgrpt,phiofr
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /modelpath/ k,rref,modeltype

```

```

COMMON /parampath/ p0,G,m0overM0
COMMON /lgrpath/ lgr
COMMON /phiandr/ lgphi,dlgphidlgr,d2lgphidlgr2,spd2phidr2,
*      spd3phidr3,spd4phidr4,spd2rdphi2
r = 1.d1**lgrpt
if (modeltype.eq.1) then ! polytropes
  if (lgrpt.lt.lgr(1)) then ! small r asymptotes
    d2phidr2ofr = p0/3.d0 -
*      k*(p0**2.d0)*(r**2.d0)/(2.d1*(k-1.d0)) +
*      k*(3.d0*k+1.d1)*(p0**3.d0)*(r**4.d0)/((2016.d0)
*      *((k-1.d0)**2.d0)) -
*      k*(9.d0*(k**2.d0)+8.6d1*k+2.8d2)*(p0**4.d0)*
*      (r**6.d0)/(46656.d1*((k-1.d0)**3.d0))
  else if (lgrpt.gt.lgr(modnum)) then ! large r asymptotes
    if (k.lt.2.d0) then
      d2phidr2ofr = -((P0/(3.d0-k))*(5.d-1*(2.d0-k)))*
*      ((2.d0*(k-1.d0))*(5.d-1*k))*(1.d1*(-k*lgrpt))/(k-1.d0)
    else if (k.gt.2.d0) then
      d2phidr2ofr = -2.d0*((P0/(2.d0*(k-1.d0)*(3.d0-k))
*      *(5.d-1*(2.d0-k)))*(1.d1*(-k*lgrpt))
    else
      d2phidr2ofr = -2.d0*(1.d1*(-2.d0*lgrpt))
    end if
  else ! numerical results for inbetween bits
    splintcall = 'd2phidr2ofr_lgr_d2lgphidlgr2'
    CALL splintcheck(lgr,d2lgphidlgr2,spd4phidr4,modnum,lgrpt,
*      d2lgphidlgr2pt,.false.)
    splintcall = 'd2phidr2ofr_lgr_dlgphidlgr'
    CALL splintcheck(lgr,dlgphidlgr,spd3phidr3,modnum,lgrpt,
*      dlgphidlgrpt,.false.)
    d2phidr2ofr = (1.d1**log10(phiofr(lgrpt))-2.d0*lgrpt))*
*      (dlgphidlgrpt*(dlgphidlgrpt-1.d0)+d2lgphidlgr2pt/log(1.d1))
  end if
  else if (modeltype.eq.2) then ! king models
    if (lgrpt.lt.lgr(1)) then ! small r asymptotes
      d2phidr2ofr = p0/3.d0 + (r**2.d0)*(p0**2.d0)
*      *(2.d0*((k/PI)**5.d-1)-exp(k)*error(k**5.d-1))/1.d1
    else ! numerical results for everything between small r and r_t
      splintcall = 'd2phidr2ofr_lgr_d2lgphidlgr2'
      CALL splintcheck(lgr,d2lgphidlgr2,spd4phidr4,modnum,lgrpt,
*      d2lgphidlgr2pt,.false.)
      splintcall = 'd2phidr2ofr_lgr_dlgphidlgr'
      CALL splintcheck(lgr,dlgphidlgr,spd3phidr3,modnum,lgrpt,
*      dlgphidlgrpt,.false.)
      d2phidr2ofr = (1.d1**log10(phiofr(lgrpt))-2.d0*lgrpt))*
*      (dlgphidlgrpt*(dlgphidlgrpt-1.d0)+d2lgphidlgr2pt/log(1.d1))
    end if
  else if (modeltype.eq.3) then ! dehn models
    d2phidr2ofr = (2.d0**((4.d0-k))*p0*((r+1.d0)**(k-4.d0)))*
*      (r**(-k))*(1.d0-k-2.d0*r)/(3.d0-k)
  else if (modeltype.eq.4) then ! powerlaw models
    d2phidr2ofr = p0*(1.d0-k)*(r**(-k))/(3.d0-k)
  end if
  return
END

! squared angular velocity of a circular orbit with radius r
FUNCTION omegac2ofr(lgrpt)
  IMPLICIT NONE
  INTEGER modnum,modeltype
  PARAMETER (modnum=1000)
  REAL*8 k,rref,omegac2ofr,lgrpt,lgr(modnum),lgomegac2(modnum)
  REAL*8 spd2omegac2dr2(modnum),lgomegac2pt
  CHARACTER*30 splintcall
  COMMON /splintpath/ splintcall
  COMMON /modelpath/ k,rref,modeltype
  COMMON /lgrpath/ lgr
  COMMON /omegacpath/ lgomegac2,spd2omegac2dr2
  splintcall = 'omegac2ofr_lgr_lgomegac2'
  CALL splintcheck(lgr,lgomegac2,spd2omegac2dr2,modnum,lgrpt,
*      lgomegac2pt,.false.)
  omegac2ofr = 1.d1**lgomegac2pt
  return
END

! density of host galaxy at radius r
FUNCTION rhoofr(lgrpt)
  IMPLICIT NONE
  INTEGER modnum,modeltype
  PARAMETER (modnum=1000)
  REAL*8 k,rref,rhoofr,lgrpt,lgr(modnum),lgrho(modnum),lgrhopt
  REAL*8 spd2rhodr2(modnum)
  CHARACTER*30 splintcall

```

```

COMMON /splintpath/ splintcall
COMMON /modelpath/ k,rref,modeltype
COMMON /lgrpath/ lgr
COMMON /rhopath/ lgrho,spd2rhodr2
splintcall = 'rhoofr_lgr_lgrho'
CALL splintcheck(lgr,lgrho,spd2rhodr2,modnum,lgrpt,
*          lgrhopt, .false.)
rhoofr = 1.d1**lgrhopt
return
END

! first derivative of host galaxy density w.r.t. radius
FUNCTION drhodr(lgrpt)
IMPLICIT NONE
REAL*8 PI
PARAMETER (PI=3.141592654d0)
INTEGER modeltype
REAL*8 drhodr,lgrpt,k,rref,phiofr,drhodphi,dphidrofr,W,error
COMMON /modelpath/ k,rref,modeltype
if (modeltype.eq.1) then
drhodphi = -(k/(2.d0*(k-1.d0)))*(1.d0-((k-2.d0)/(2.d0*(k-1.d0)))
*          *phiofr(lgrpt))*k/(k-2.d0))
drhodphi = drhodphi*dphidrofr(lgrpt)
else if (modeltype.eq.2) then
W = k-phiofr(lgrpt)
drhodphi = -(exp(W)*error(W**5.d-1)-2.d0*((W/PI)**5.d-1)/
*          (exp(k)*error(k**5.d-1)-((4.d0*k/PI)**5.d-1)*
*          (1.d0+2.d0*k/3.d0)))
drhodphi = drhodphi*dphidrofr(lgrpt)
else if (modeltype.eq.3) then
drhodphi = -(2.d0**((4.d0-k))*(1.d1**(-lgrpt*(k+1.d0)))*
*          ((1.d1**lgrpt)+1.d0)**(k-5.d0))*(4.d0*(1.d1**lgrpt)+k)
else if (modeltype.eq.4) then
drhodphi = -k*(1.d1**(-lgrpt*(k+1.d0)))
end if
return
END

! a modified version of sort from Press et al. 1992
! sorts multiple arrays simultaneously
SUBROUTINE multisort(numvec,n,sort1,sort2,sort3,sort4,sort5,sort6
*,sort7,sort8)
implicit none
INTEGER numvec,n,iwksp(n)
REAL*8 sort1(n,3),sort2(n,3),sort3(n,3),sort4(n,3),sort5(n,3)
REAL*8 sort6(n,3),sort7(n,3),sort8(n,3),WKSP(n)
INTEGER j,numdone

numdone=0

call indexx(n,sort1,iwksp)
do 11 j=1,n
wksp(j)=sort1(j,1)
11 continue
do 12 j=1,n
sort1(j,1)=wksp(iwksp(j))
12 continue
numdone=numdone+1
if (numdone.eq.numvec) RETURN

do 13 j=1,n
wksp(j)=sort2(j,1)
13 continue
do 14 j=1,n
sort2(j,1)=wksp(iwksp(j))
14 continue
numdone=numdone+1
if (numdone.eq.numvec) RETURN

do 15 j=1,n
wksp(j)=sort3(j,1)
15 continue
do 16 j=1,n
sort3(j,1)=wksp(iwksp(j))
16 continue
numdone=numdone+1
if (numdone.eq.numvec) RETURN

do 17 j=1,n
wksp(j)=sort4(j,1)
17 continue
do 18 j=1,n
sort4(j,1)=wksp(iwksp(j))

```

```

18      continue
      numdone=numdone+1
      if (numdone.eq.numvec) RETURN

      do 19 j=1,n
         wksp(j)=sort5(j,1)
19      continue
      do 20 j=1,n
         sort5(j,1)=wksp(iwksp(j))
20      continue
      numdone=numdone+1
      if (numdone.eq.numvec) RETURN

      do 21 j=1,n
         wksp(j)=sort6(j,1)
21      continue
      do 22 j=1,n
         sort6(j,1)=wksp(iwksp(j))
22      continue
      numdone=numdone+1
      if (numdone.eq.numvec) RETURN

      do 23 j=1,n
         wksp(j)=sort7(j,1)
23      continue
      do 24 j=1,n
         sort7(j,1)=wksp(iwksp(j))
24      continue
      numdone=numdone+1
      if (numdone.eq.numvec) RETURN

      do 25 j=1,n
         wksp(j)=sort8(j,1)
25      continue
      do 26 j=1,n
         sort8(j,1)=wksp(iwksp(j))
26      continue

      RETURN
      end

      ! slightly modifies values to force monotonicity
      SUBROUTINE monotonic(x1,n,x2,h)
      IMPLICIT NONE
      INTEGER n,i
      REAL*8 x1(n),x2(n),h
      x2(1)=x1(1)
      do i = 1,n-1
         x2(i+1)=x1(i+1)
         if ((1.d0+h)*x2(i).gt.x2(i+1)) then
            x2(i+1) = (1.d0+h)*x2(i)
         end if
      end do
      return
      END

      ! checks for extrapolation before a splint call
      SUBROUTINE splintcheck(xa,ya,y2a,n,x,y,linintrp)
      IMPLICIT NONE
      INTEGER n,splinterrn
      REAL*8 xa(n),ya(n),y2a(n),x,y
      LOGICAL linintrp
      CHARACTER*30 splintcall
      COMMON /splintpath/ splintcall
      COMMON /errpath/ splinterrn
      CALL splint(xa,ya,y2a,n,x,y,linintrp)
      if (x.lt.xa(1).or.x.gt.xa(n)) then
         splinterrn = splinterrn + 1
         if (x.lt.xa(1)) then
            write(31,11) 'below ', 'x=',x, 'xa=',xa(1), 'xa-x=',xa(1)-x,
*                               splintcall
         else if (x.gt.xa(n)) then
            write(31,11) 'above ', 'x=',x, 'xa=',xa(n), 'x-xa=',x-xa(n),
*                               splintcall
         else
            pause 'splintcheck_error'
         end if
      end if
      return
11      FORMAT(A,2X,3(A,2X,E30.15,2X),A)
      END

      ! checks for extrapolation before a splint call

```

```

SUBROUTINE splint(xa,ya,y2a,n,x,y,linintrp)
IMPLICIT NONE
INTEGER n
REAL*8 x,y,xa(n),y2a(n),ya(n)
INTEGER k,khi,klo
REAL*8 a,b,h
LOGICAL linintrp,linin
CHARACTER*30 splintcall
COMMON /splintpath/ splintcall
COMMON /linpath/ linin
klo=1
khi=n
1  if (khi-klo.gt.1) then
      k=(khi+klo)/2
      if (xa(k).gt.x) then
          khi=k
      else
          klo=k
      endif
      goto 1
    endif
h=xa(khi)-xa(klo)
if (h.eq.0.d0) then
    print *, splintcall
    do k = 1,n
        print *, xa(k),ya(k),k
    end do
    pause 'bad_xa_input_in_splint'
end if
a=(xa(khi)-x)/h
b=(x-xa(klo))/h
y=a*ya(klo)+b*ya(khi)
! if linintrp = true, interpolate linearly, otherwise do cubic interpolation
if ((linintrp.eqv..false.).and.(linin.eqv..false.)) then
    y=y+((a**3-a)*y2a(klo)+(b**3-b)*y2a(khi))*(h**2)/6.d0
end if
return
END

! the following is a slightly modified version of the driver for Runge-Kutta
! integration from Press et al. 1992, that forces a step when a step cannot
! be taken while satisfying the required eps error
SUBROUTINE odeint(ystart,nvar,x1,x2,eps,h1,hadv,nok,nbad,derivs,
*rkqs)
implicit none
INTEGER nbad,nok,nvar,KMAXX,MAXSTP,NMAX
REAL*8 eps,h1,hadv,x1,x2,ystart(nvar),TINY
EXTERNAL derivs,rkqs
PARAMETER (MAXSTP=100000000,NMAX=50,KMAXX=100,TINY=1.d-30)
INTEGER i,kmax,kount,nstp,hount
REAL*8 dxsav,h,hdid,hnext,x,xsav,dydx(NMAX),xp(KMAXX),y(NMAX),
*yp(NMAX,KMAXX),yscal(NMAX)
COMMON /path/ kmax,kount,dxsav,xp,yp,hount
x = x1
h = sign(h1,x2-x1)
hadv = sign(hadv,x2-x1) ! align sign of hadv with direction in which
! integration is proceeding

nok = 0
nbad = 0
kount = 0
hount = 0
do 11 i=1,nvar
    y(i)=ystart(i)
11  continue
if (kmax.gt.0) xsav=x-2.d0*dxsav
do 16 nstp=1,MAXSTP
    call derivs(x,y,dydx)
    do 12 i=1,nvar
        yscal(i)=abs(y(i))+abs(h*dydx(i))+TINY
12  continue
if (kmax.gt.0) then
    if (abs(x-xsav).gt.abs(dxsav)) then
        if (kount.lt.kmax-1) then
            kount=kount+1
            xp(kount)=x
            do 13 i=1,nvar
                yp(i,kount)=y(i)
13  continue
            xsav=x
        endif
    endif
endif
if ((x+h-x2)*(x+h-x1).gt.0.d0) h=x2-x

```

```

      call rkqs(y,dydx,nvar,x,h,hadv,eps,yscal,hdid,hnext,derivs)
      if (hdid.eq.hadv) hount = hount+1 ! track number of forced advances
                                         ! made
      if (hdid.eq.h) then
        nok=nok+1
      else
        nbad=nbad+1
      endif
      if ((x-x2)*(x2-x1).ge.0.d0) then
        do 14 i=1,nvar
          ystart(i)=y(i)
14      continue
          if (kmax.ne.0) then
            kount=kount+1
            xp(kount)=x
            do 15 i=1,nvar
              yp(i,kount)=y(i)
15      continue
            endif
            return
          endif
        h=hnext
16      continue
        pause 'too many steps in codeint'
33      return
      END

      ! the following is a slightly modified version of the 'quality step'
      ! routine from Press et al. 1992 to enable aborting the integration
      ! of orbits that have become unphysical
      SUBROUTINE rkqs(y,dydx,n,x,htry,hadv,eps,yscal,hdid,hnext,derivs)
      implicit none
      INTEGER n,NMAX
      REAL*8 eps,hdid,hnext,htry,x,dydx(n),y(n),yscal(n),hadv
      EXTERNAL derivs
      PARAMETER (NMAX=50)
      CU      USES derivs,rkck
      INTEGER i
      REAL*8 errmax,h,htemp,xnew,yerr(NMAX),ytemp(NMAX),SAFETY,PGROW,
      *PSHRNK,ERRCON,tf
      LOGICAL notforce,abort
      PARAMETER(SAFETY=.9d0,PGROW=-.2d0,PSHRNK=-.25d0,ERRCON=1.89d-4)
      common /abortpath/ tf,abort
      abort = .false.
      notforce = .true.
      h=htry
1      call rkck(y,dydx,n,x,h,ytemp,yerr,derivs)
      if (abort) then
        do i = 1,n
          y(i) = -4.d1
        end do
        x = log10(tf)
        return
      end if
      errmax=0.d0
      do 11 i=1,n
        errmax=max(errmax,abs(yerr(i)/yscal(i)))
11      continue
      errmax=errmax/eps
2      if (errmax.gt.1.d0.and.notforce) then
        htemp=SAFETY*h*(errmax**PSHRNK)
        h=sign(max(abs(htemp),1.d-1*abs(h)),h)
        xnew=x+h
        if (xnew.eq.x) then ! if new step is the same as the previous failed
                           ! step, force a step of hadv
          h = hadv
          notforce = .false.
        end if
        goto 1
      else
        if (errmax.gt.ERRCON) then
          hnext=SAFETY*h*(errmax**PGROW)
        else
          hnext=5.d0*h
        endif
        hdid=h
        x=x+h
        do 12 i=1,n
          y(i)=ytemp(i)
12      continue
        return
      endif
      END

```



```
FUNCTION rtbis (func , x1 , x2 , xacc)
SUBROUTINE rkck (y , dydx , n , x , h , yout , yerr , derivs )
FUNCTION error (x)
FUNCTION gam (xx)
FUNCTION gammp (a , x)
SUBROUTINE gcf (gammcf , a , x , gln)
SUBROUTINE gser (gamser , a , x , gln)
SUBROUTINE spline (x , y , n , ypl , ypn , y2)
SUBROUTINE indexx (n , arr , indx)
FUNCTION ran2 (idum)
```

Bibliography

- Aarseth S. J., Heggie D. C., 1993, *Astronomical Society of the Pacific Conference Series*, 48, 701
- Aguilar L., Hut P., Ostriker J. P., 1988, *ApJ*, 335, 720
- Ambartsumian V. A., 1938, in Goodman J., Hut P., eds, *On the dynamics of open clusters*, *Uch. Zap. LGU* 22, 19; transl. in *Dynamics of Star Clusters*, 1985, IAU Symposium, p. 521
- Anders P., de Grijs R., Fritze-v. Alvensleben U., Bissantz N., 2004, *MNRAS*, 347, 17
- Andreuzzi G., Buonanno R., Fusi Pecci F., Iannicola G., Marconi G., 2000, *A&A*, 353, 954
- Bahcall J. N., Ostriker J. P., 1975, *Nature*, 256, 23
- Bahcall J. N., Wolf R. A., 1976, *ApJ*, 209, 214
- Barmby P., McLaughlin D. E., Harris W. E., Harris G. L. H., Forbes D. A., 2007, *AJ*, 133, 2764
- Barmby P., Perina S., Bellazzini M., Cohen J. G., Hodge P. W., Huchra J. P., Kissler-Patig M., Puzia T. H., Strader J., 2009, *AJ*, 138, 1667
- Barmby P., Holland S., Huchra J. P., 2002, *AJ*, 123, 1937
- Barmby P., Huchra J. P., Brodie J. P., 2001, *AJ*, 121, 1482
- Bastian N., Emsellem E., Kissler-Patig M., Maraston C., 2006, *A&A*, 445, 471
- Bastian N., Ercolano B., Gieles M., Rosolowsky E., Scheepmaker R. A., Gutermuth R., Efremov Y., 2007, *MNRAS*, 379, 1302
- Bastian N., Schweizer F., Goudfrooij P., Larsen S. S., Kissler-Patig M., 2013, *MNRAS*, 431, 1252
- Baumgardt H., Makino J., 2003, *MNRAS*, 340, 227
- Baumgardt H., 1998, *A&A*, 330, 480
- Baumgardt H., 2001, *MNRAS*, 325, 1323
- Baumgardt H., Parmentier G., Gieles M., Vesperini E., 2010, *MNRAS*, 401, 1832

- Baumgardt H., Kroupa P., Parmentier G., 2008, MNRAS, 384, 1231
- Bedin L. R., Piotto G., Anderson J., Cassisi S., King I. R., Momany Y., Carraro G., 2004, ApJ, 605, 125
- Bekki K., Beasley M. A., Brodie J. P., Forbes D. A., 2005, MNRAS, 363, 1211
- Binney J., Tremaine S., 2008, Galactic Dynamics, Princeton University Press, 2nd edition
- Bontekoe T. R., van Albada T. S., 1987, MNRAS, 224, 349
- Brosche P., Odenkirchen M., Geffert M., 1999, New Astronomy, 4, 133
- Bruzual G., Charlot S., 2003, MNRAS, 344, 1000
- Caputo F., Castellani V., 1984, MNRAS, 207, 185
- Chabrier G., 2003, PASP, 115, 763
- Chandar R., Puzia T. H., Sarajedini A., Goudfrooij P., 2006, ApJ, 646, 107
- Chandar R., Whitmore B. C., Kim H., Kaleida C., Mutchler M., Calzetti D., Saha A., O'Connell R., Balick B., Bond H., Carollo M., Disney M., Dopita M. A., Frogel J. A., Hall D., Holtzman J. A., Kimble R. A., McCarthy P., Paresce R., Silk J., Trauger J., Walker A. R., Windhorst R. A., Young E., 2010, ApJ, 719, 966
- Chandar R., Fall M. S., McLaughlin D. E., 2007, ApJ, 668, 119
- Chandrasekhar S., 1942, Principles of Stellar Dynamics, The University of Chicago Press, 1st edition
- Chandrasekhar S., 1943, ApJ, 97, 255
- Chernoff D. F., Djorgovski S., 1989, ApJ, 339, 904
- Chernoff D. F., Weinberg M. D., 1990, ApJ, 351, 121
- Chernoff D. F., Kochanek C. S., Shapiro S. L., 1986, ApJ, 309, 183
- Cora S. A., Vergne M. M., Muzzio J. C., 1997, MNRAS, 289, 253
- Cora S. A., Vergne M. M., Muzzio J. C., 2001, ApJ, 456, 165
- Côté P., 1999, AJ, 118, 406
- Côté P., McLaughlin D. E., Hanes D. A., Bridges T. J., Geisler D., Merritt D., Hesser J. E., Harris G. L. H., Lee M. G., 2001, ApJ, 559, 828

- Côté P., McLaughlin D. E., Cohen J. G., Blakeslee J. P., 2003, *ApJ*, 591, 850
- Côté P., Blakeslee J. P., Ferrarese L., Jordán A., Mei S., Merritt D., Milosavljević M., Peng E. W., Tonry J. L., West M. J., 2004, *ApJS*, 153, 223
- de Grijs R., Anders P., Bastian N., Lynds R., Lamers H. J. G. L. M., O’Neil E. J., 2003, *MNRAS*, 343, 1285
- de La Fuente Marcos R., 1995, *A&A*, 301, 407
- Dehnen W., 1993, *MNRAS*, 265, 250
- Dekel A., Devor J., Hetzroni G., 2003, *MNRAS*, 341, 326
- Djorgovski S., Davis M., 1987, *ApJ*, 313, 59
- Djorgovski S., Meylan G., 1994, *AJ*, 108, 1292
- Djorgovski S., 1995, *ApJ*, 438, 29
- Dressler A., Lynden-Bell D., Burstein D., Davies R. L., Faber S. M., Terlevich R., Wegner G., 1987, *ApJ*, 313, 42
- Dubath P., Grillmair C. J., 1997, *A&A*, 321, 379
- Durrell P. R., Harris W. E., Geisler D., Pudritz R. E., 1996, *AJ*, 112, 972
- Elson R. A. W., Fall M. S., Freeman K. C., 1987, *ApJ*, 323, 54
- Faber S. M., Gallagher J. S., 1979, *Annu. Rev. Astron. Astrophys.*, 17, 135
- Fabio A., Merritt D., 2012, *ApJ*, 745, 83
- Fall S. M., Rees M. J., 1977, *MNRAS*, 181, 37
- Fall S. M., Rees M. J., 1985, *ApJ*, 298, 18
- Fall S. M., Rees M. J., 1988, in Grindlay J., Philip A., eds, *The Harlow-Shapley Symposium on Globular Cluster Systems in Galaxies*, Kluwer Academic Publishers, p. 323
- Fall M. S., Zhang Q., 2001, *ApJ*, 561, 751
- Fall S. M., Chandar R., Whitmore B. C., 2005, *ApJ*, 631, 133
- Ferraro F. R., Carretta E., Bragaglia A., Renzini A., Ortolani S., 1997, *MNRAS*, 286, 1012

- Fioc M., Rocca-Volmerange B., 1997, *A&A*, 326, 950
- Frenk C. S., White S. D. M., 1980, *MNRAS*, 193, 295
- Fukushige T., Heggie D. C., 2000, *MNRAS*, 318, 753
- G. C. J., Ryzhov A., 1997, *ApJ*, 486, 230
- Gerhard O. E., 1991, *MNRAS*, 250, 812
- Gerhard O. E., 1993, *MNRAS*, 265, 213
- Gerhard O. E., Jeske G., Saglia R. P., Bender R., 1998, *MNRAS*, 295, 197
- Gieles M., Baumgardt H., 2008, *MNRAS*, 389, 28
- Gieles M., Baumgardt H., Heggie D. C., 2010, *MNRAS*, 408, 16
- Gieles M., Larsen S. S., Scheepmaker R. A., Bastian N., Haas M. R., Lamers H. J. G. L. M., 2006, *A&A*, 446, 9
- Gieles M., Heggie D. C., Zhao H., Lamers H. J. G. L. M., 2011, *MNRAS*, 413, 2509
- Giersz M., Heggie D. C., 1994, *MNRAS*, 268, 257
- Giersz M., Heggie D. C., 1996, *MNRAS*, 279, 1037
- Giersz M., Heggie D. C., 1997, *MNRAS*, 286, 709
- Giersz M., 2001, *MNRAS*, 324, 218
- Glatt K., Grebel E. K., Jordi K., Gallagher III J. S., Da Costa G., Clementini G., Tosi M., Harbeck D., Nota A., Sabbi E., Sirianni M., 2011, *AJ*, 142, 36
- Gnedin O. Y., Ostriker J. P., 1997, *ApJ*, 474, 223
- Gnedin O. Y., Hernquist L., Ostriker J. P., 1999, *ApJ*, 514, 109
- Gnedin O. Y., Lee H. M., Ostriker J. P., 1999, *ApJ*, 522, 935
- Gonzaga S., 2011, *ACS Data Handbook*, Baltimore: STScI, 6.0 edition
- Goodman J., 1987, *ApJ*, 313, 576
- Goudfrooij P., 2012, *ApJ*, 750, 140
- Goudfrooij P., Gilmore D., Whitmore B. C., Schweizer F., 2004, *ApJ*, 613, 121

- Grillmair C. J., C. F. K., V. B. G., Carter D., Couch W. J., Sommer-Larsen J., Taylor K., 1994, *ApJ*, 422, 9
- Gunn J. E., Griffin R. F., 1979, *AJ*, 84, 752
- Hanes D. A., 1977, *MNRAS*, 180, 309
- Harris W. E., 1991, *Annu. Rev. Astron. Astrophys.*, 29, 543
- Harris W. E., 1996, *AJ*, 112, 1487
- Harris W. E., Kavelaars J. J., Hanes D. A., Pritchett C. J., Baum W. A., 2009, *AJ*, 137, 3314
- Harris E. W., Harris G. L. H., McLaughlin D. E., 1998, *AJ*, 155, 1801
- Heggie D. C., Hut P., 2003, *The Gravitational Million-Body Problem*, Cambridge University Press, 1st edition
- Heggie D. C., 1975, *MNRAS*, 173, 729
- Heggie D. C., 1979, *MNRAS*, 118, 525
- Heggie D. C., Giersz M., Spurzem R., Takahashi K., 1998, *Highlights in Astronomy*, 11, 551
- Hénon M., 1961, *Annales d'Astrophysique*, 24, 369
- Hénon M., 1965, *Annales d'Astrophysique*, 28, 62
- Hernquist L., 1990, *ApJ*, 356, 359
- Heyl J., 2007, *MNRAS*, 381, 70
- Hills J. G., 1975, *AJ*, 80, 809
- Hills J. G., 1980, *ApJ*, 235, 986
- Holtzman J. A., Faber S. M., Shaya E. J., Lauer T. R., Groth J., Hunter D. A., Baum W. A., Ewald S. P., Hester J. J., Light R. M., Lynds C. R., O'Neil E. J., Westphal J. A., 1992, *AJ*, 103, 691
- Howell J. H., Guhathakurta P., Tan A., 2000, *AJ*, 119, 1259
- Hurley J. R., Mackey A. D., 2010, *MNRAS*, 408, 2353
- Hurley J. R., Pols O. R., Tout C. A., 2000, *MNRAS*, 315, 543

- Hut P., 1986, in Unknown, ed., The role of binaries in globular cluster evolution, The evolution of galactic X-ray binaries, p. 1
- Iben I., Renzini A., 1983, *Annu. Rev. Astron. Astrophys.*, 21, 271
- Innanen K. A., Harris W. E., Webbink R. F., 1983, *AJ*, 88, 338
- Inoue S., 2009, *MNRAS*, 397, 709
- Jaffe W., 1983, *MNRAS*, 202, 995
- Jordán A., Blakeslee J. P., Peng E. W., Mei S., Côté P., Ferrarese L., Tonry J. L., Merritt D., Milosavljević M., West M. J., 2004, *ApJ*, 154, 509
- Jordán A., Côté P., Blakeslee J. P., Ferrarese L., McLaughlin D. E., Mei S., Peng E. W., Tonry J. L., Merritt D., Milosavljević M., Sarazin C. L., Sivakoff G. R., West M. J., 2005, *ApJ*, 634, 1002
- Jordán A., McLaughlin D. E., Côté P., Ferrarese L., Peng E. W., Blakeslee J. P., Mei S., Villegas D., Merritt D., Tonry J. L., West M. J., 2006, *ApJ*, 651, 25
- Jordán A., McLaughlin D. E., Côté P., Ferrarese L., Peng E. W., Mei S., Villegas D., Merritt D., Tonry J. L., West M. J., 2007, *ApJ*, 171, 101
- Jordán A., Peng E. W., Blakeslee J. P., Côté P., Eyheramendy S., Ferrarese L., Mei S., Tonry J. L., West M. J., 2009, *ApJS*, 180, 54
- Just A., Peñarrubia J., 2005, *A&A*, 431, 861
- Just A., Khan F. M., Berczik P., Ernst A., Spurzem R., 2011, *MNRAS*, 411, 653
- Kafle P. R., Sharma S., Lewis G. F., Bland-Hawthorn J., 2013, *MNRAS*, 430, 2973
- Kazantzidis S., Magorrian J., Moore B., 2004, *ApJ*, 601, 37
- King I. R., 1959, *AJ*, 64, 351
- King I. R., 1962, *AJ*, 67, 471
- King I. R., 1966, *AJ*, 71, 64
- King I. R., Sosin C., Cool A. M., 1995, *ApJ*, 452, 33
- Koch A., Grebel E. K., Odenkirchen M., Martínez-Delgado D., Caldwell J. A. R., 2004, *AJ*, 128, 2274

- Kong A. K. H., Heinke C. O., di Stefano R., Cohn H. N., Lugger P. M., Barmby P., Lewin W. H. G., Primi F. A., 2010, MNRAS, 407, 84
- Kravtsov A. V., Gnedin O. Y., Klypin A. A., 2004, ApJ, 609, 482
- Kroupa P., 2001, MNRAS, 322, 231
- Kruijssen J. M. D., Lamers H. J. G. L. M., 2008, A&A, 490, 151
- Kundic T., Ostriker J. P., 1995, ApJ, 438, 702
- Küpper A. H. W., Kroupa P., Baumgardt H., Heggie D. C., 2010, MNRAS, 407, 2241
- Küpper A. H. W., Kroupa P., Baumgardt H., 2008, MNRAS, 389, 889
- Küpper A. H. W., Lane R. R., Heggie D. C., 2012, MNRAS, 420, 2700
- Lada C. J., Lada E. A., 2003, Annu. Rev. Astron. Astrophys., 41, 57
- Lamers H. J. G. L. M., Baumgardt H., Gieles M., 2010, MNRAS, 409, 305
- Lamers H. J. G. L. M., Baumgardt H., Gieles M., 2013, MNRAS, 433, 1378
- Larsen S. S., Richtler T., 1999, A&A, 345, 59
- Larsen S. S., Richtler T., 2000, A&A, 354, 836
- Larsen S. S., 2002a, AJ, 124, 1393
- Larsen S. S., 2002b, in Geisler D., Grebel E., Minniti D., eds, Open, Massive and Globular Clusters – Part of the Same Family?, IAU Symposium, p. 421
- Larson R. B., 1993, in Smith G. H., Brodie J. P., eds, Present-Day Cluster Formation, Astronomical Society of the Pacific Conference Series, p. 675
- Lee H. M., Goodman J., 1995, ApJ, 443, 109
- Lee H. M., Ostriker J., 1987, ApJ, 322, 123
- Lee H. M., 1987, ApJ, 319, 801
- Lee M. G., Hwang H. S., Kim S. C., Park H. S., Geisler D., Sarajedini A., Harris W. E., 2008, ApJ, 674, 886
- Lee H. M., Fahlman G. G., Richer H. B., 1991, ApJ, 366, 455
- Lynden-Bell D., Wood R., 1968, MNRAS, 138, 495

- Makino J., 1996, *ApJ*, 471, 796
- Maraston C., 2005, *MNRAS*, 362, 799
- Marchant A. B., Shapiro S. L., 1980, *ApJ*, 239, 685
- McLaughlin D. E., Fall M. S., 2008, *ApJ*, 679, 1272
- McLaughlin D. E., van der Marel R. P., 2005, *ApJ*, 161, 304
- McLaughlin D. E., 1999, *AJ*, 117, 2398
- McLaughlin D. E., Barmby P., Harris W. E., Forbes D. A., Harris G. L. H., 2008, *MNRAS*, 384, 563
- Mei S., Blakeslee J. P., Tonry J. L., Jordán A., Peng E. W., Côté P., Ferrarese L., West M. J., Merritt D., Milosavljević M., 2005, *ApJ*, 625, 121
- Mei S., Blakeslee J. P., Côté P., Tonry J. L., West M. J., Ferrarese L., Jordán A., Peng E. W., Anthony A., Merritt D., 2007, *ApJ*, 655, 144
- Merritt D., 1985, *MNRAS*, 214, 25
- Meurer G. R., 2000, in Lançon A., Boily C., eds, *Star Clusters and the Duration of Starbursts*, *Astronomical Society of the Pacific Conference Series*, p. 81
- Meylan G., 1987, *A&A*, 184, 144
- Michie R. W., Bodenheimer P. H., 1963, *MNRAS*, 126, 269
- Miller G. E., Scalo J. M., 1979, *ApJS*, 41, 513
- Miller B. W., Lotz J. M., Ferguson H. C., Stiavelli M., Whitmore B. C., 1998, *ApJ*, 508, 133
- Minniti D., Kissler-Patig M., Goudfrooij P., Meylan G., 1998, *AJ*, 115, 121
- Murali C., Weinberg M. D., 1997, *MNRAS*, 291, 717
- Odenkirchen M., Brosche P., Geffert M., Tucholke H.-J., 1997, *New Astronomy*, 2, 4770
- Okazaki T., Tosa M., 1995, *MNRAS*, 274, 480
- Osipkov L. P., 1979, *Soviet Astronomy Letters*, 5, 42
- Ostriker J. P., Gnedin O. Y., 1997, *ApJ*, 487, 667
- Parmentier G., Gilmore G., 2007, *MNRAS*, 377, 352

- Peebles P. J. E., Dicke R. H., 1968, *ApJ*, 154, 891
- Peng E. W., Jordán A., Côté P., Takamiya M., West M. J., Blakeslee J. P., Chen C., Ferrarese L., Mei S., Tonry J. L., West A. A., 2008, *ApJ*, 681, 197
- Pepe C., Pellizza L. J., 2013, *MNRAS*, 430, 2789
- Piotto G., Bedin L. R., Anderson J., King I. R., Cassisi S., Milone A. P., Villanova S., Pietrinferni A., Renzini A., 2007, *ApJ*, 661, 53
- Piskunov A. E., Kharchenko N. V., Röser S., Schilbach E., Scholz R.-D., 2006, *A&A*, 445, 545
- Plummer H. C., 1911, *MNRAS*, 71, 460
- Pota V., Forbes D. A., Romanowsky A. J., Brodie J. P., Spitler L. R., Strader J., Foster C., Arnold J. A., Benson A., Blom C., Hargis J. R., Rhode K. L., Usher C., 2013, *MNRAS*, 428, 389
- Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., 1992, *Numerical Recipes in Fortran*, Cambridge University Press, 2nd edition
- Prieto J. L., Gnedin O. Y., 2008, *ApJ*, 689, 919
- Read J. I., Geordt T., Moore B., Pontzen A. P., Stadel J., Lake G., 2006a, *MNRAS*, 373, 1451
- Read J. I., Wilkinson M. I., Evans N. W., Gilmore G., Kleya J. T., 2006b, *MNRAS*, 366, 429
- Renaud F., Boily C. M., Naab T., Theis C., 2009, *ApJ*, 706, 67
- Schechter P., 1976, *ApJ*, 203, 297
- Schulz J., Fritze-v. Alvensleben U., M'oller C. S., Fricke K. J., 2002, *A&A*, 392, 1
- Schweizer F., 1986, *Science*, 231, 227
- Schweizer F., Miller B. W., Whitmore B. C., Fall S. M., 1996, *AJ*, 112, 1839
- Sparke L. S., Gallagher J. S., 2007, *Galaxies in the Universe, An Introduction*, Cambridge University Press, 2nd edition
- Spitzer L., Chevalier R. A., 1973, *ApJ*, 183, 565
- Spitzer, Lyman J., Härm R., 1958, *ApJ*, 127, 544

- Spitzer L., Shull J. M., 1975, ApJ, 201, 773
- Spitzer L., Thuan T. X., 1972, ApJ, 175, 31
- Spitzer L., 1969, ApJ, 158, 139
- Spitzer L., 1987, Dynamical Evolution of Globular Clusters, Princeton University Press, 1st edition
- Sugimoto D., Bettwieser E., 1983, MNRAS, 204, 19
- Takahashi K., Baumgardt H., 2012, MNRAS, 420, 1799
- Tanikawa A., Fukushige T., 2005, PASJ, 57, 155
- Tanikawa A., Fukushige T., 2010, PASJ, 62, 1215
- Trenti M., Heggie D. C., Hut P., 2007, MNRAS, 374, 344
- van den Bosch F. C., Lewis G. F., Lake G., Stadel J., 1999, ApJ, 515, 50
- Vesperini E., Heggie D. C., 1997, MNRAS, 289, 898
- Vesperini E., Zepf S. E., 2003, ApJ, 587, 97
- Vesperini E., 1998, MNRAS, 299, 1019
- Vesperini E., 2000, MNRAS, 318, 841
- Vesperini E., Zepf S. E., Kundu A., Ashman K. M., 2003, ApJ, 593, 760
- von Hoerner S., 1957, ApJ, 125, 451
- Webb J. J., Harris W. E., Sills A., Hurley J. R., 2013, ApJ, 764, 124
- Weinberg M. D., 1994a, AJ, 108, 1398
- Weinberg M. D., 1994b, AJ, 108, 1403
- Weinberg M. D., 1994c, AJ, 108, 1414
- White S. D. M., 1983, ApJ, 274, 53
- Whitmore B. C., Schweizer F., 1995, AJ, 109, 960
- Whitmore B. C., Schweizer F., Kundu A., Miller B. W., 2002, AJ, 124, 147

- Wielen R., 1988, in Grindlay J., Philip A., eds, *The Harlow-Shapley Symposium on Globular Cluster Systems in Galaxies*, Kluwer Academic Publishers, p. 393
- Woodley K. A., Gómez M., Harris W. E., Geisler D., Harris G. L. H., 2010, *AJ*, 139, 1871
- Woosley S. E., Heger A., 2007, *Physics Reports*, 442, 269
- Wu X., Tremaine S. D., 2006, *ApJ*, 643, 210
- Zaritsky D., White S. D. M., 1988, *MNRAS*, 235, 289
- Zhang Q., Fall M. S., 1999, *ApJ*, 527, 81
- Zhao H., 1997, *MNRAS*, 287, 525